PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-091704
(43)Date of publication of application : 28.03.2003

(51)Int.CI. G06K 19/07
B42D 15/10

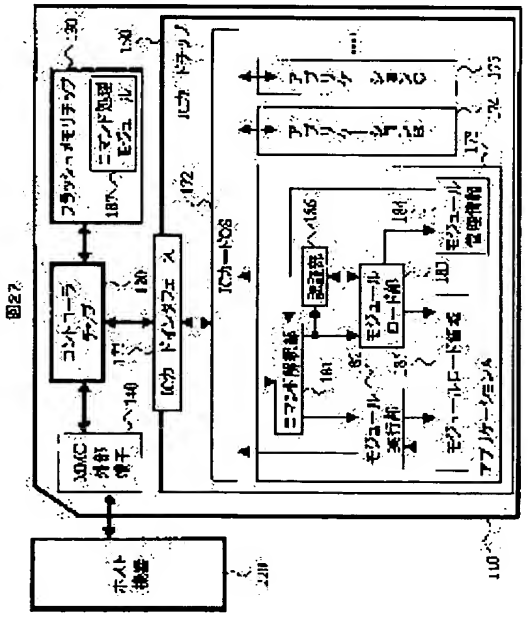
(21)Application number : 2001-388727
(22)Date of filing : 21.12.2001

(71)Applicant : HITACHI LTD
(72)Inventor : MIZUSHIMA EIGA
TSUNODA MOTOYASU
HATANO TOMIHISA
KATAYAMA KUNIHIRO
TANAKA NORIO
TSUNEHIRO TAKASHI
KIMURA KOICHI

(30)Priority
Priority number : 2001207212 Priority date : 09.07.2001 Priority country : JP

(54) STORAGE UNIT WITH NON-VOLATILE MEMORY AND INFORMATION PROCESSING DEVICE WITH FREELY DETACHABLE STORAGE UNIT

(57)Abstract:
PROBLEM TO BE SOLVED: To make an IC execute a large amount of processing even when the storage capacity in the IC (for example, ROM and EERROM) is small.
SOLUTION: This storage unit is provided with a flash memory chip 130, an IC card chip 150 capable of executing the security processing (encryption, decryption and the like), and a controller chip 12 for controlling the access to the flash memory chip and the IC card chip in accordance with the request from a host 220, a program of an IC card necessary for the security processing and a part of the data are stored in the flash memory chip, or the IC card chip requests the data processing utilizing the flash memory chip and the host equipment.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of

rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2003－91704
（Ｐ２００３－９１７０４Ａ）
(43)公開日　平成15年３月28日(2003.3.28)

(51)Int.Cl.[7]	識別記号	ＦＩ		テーマコード（参考）
G06K 19/07		B42D 15/10	521	2C005
B42D 15/10	521	G06K 19/00	N	5B035

審査請求　未請求　請求項の数18　ＯＬ　（全39頁）

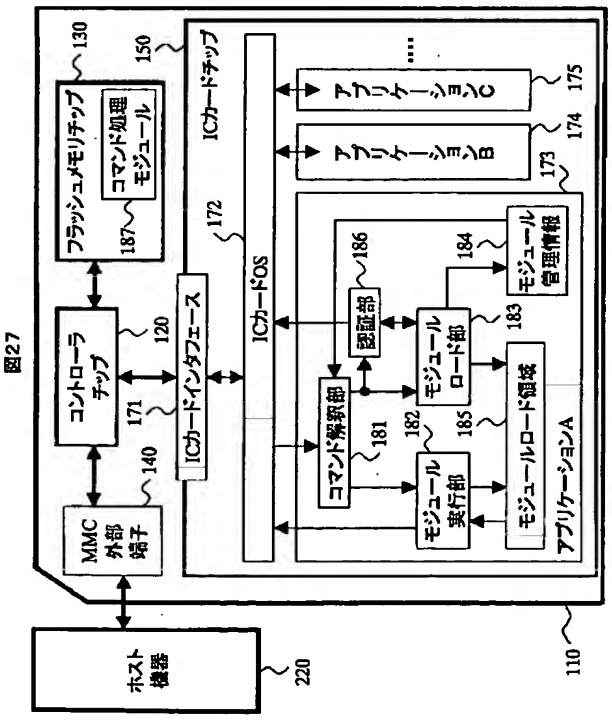
(21)出願番号　特願2001－388727（Ｐ2001－388727）
(22)出願日　平成13年12月21日(2001.12.21)
(31)優先権主張番号　特願2001－207212（Ｐ2001－207212）
(32)優先日　平成13年７月９日(2001.7.9)
(33)優先権主張国　日本（ＪＰ）

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(72)発明者　水島　永雅
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内
(72)発明者　角田　元泰
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内
(74)代理人　100075096
弁理士　作田　康夫

最終頁に続く

(54)【発明の名称】不揮発性メモリを備えた記憶装置及びその記憶装置が着脱自在な情報処理装置

(57)【要約】
【課題】本発明は、ＩＣ内の記憶容量（例えば、ＲＯＭやＥＥＲＲＯＭ、ＲＡＭ）が小さい場合にも、ＩＣに多くの処理を実行させる。
【解決手段】本発明は、フラッシュメモリチップ１３０と、セキュリティ処理（暗号化や復号化等）を実行可能なＩＣカードチップ１５０と、ホスト２２０からの要求に応じてフラッシュメモリチップ及びＩＣカードチップへのアクセスを制御するためのコントローラチップ１２０とを備え、セキュリティ処理に必要なＩＣカードのプログラムやデータの一部をフラッシュメモリチップに格納し、又は、ＩＣカードチップがフラッシュメモリチップやホスト機器を利用したデータ処理を要求する。



1

【特許請求の範囲】

【請求項１】ホスト機器からのデータを記憶するためのデータ領域を有する不揮発性メモリと、セキュリティ処理を実行するためのＩＣと、前記不揮発性メモリ及び前記ＩＣへのアクセスを制御するためのコントローラとを備えた記憶装置において、

前記不揮発性メモリは、前記アプリケーションプログラムの一部を格納するための管理領域前記ＩＣは、前記セキュリティ処理を前記ＩＣに実行させるためのアプリケーションプログラムの他部を格納するための格納領域を有する記憶装置。

【請求項２】前記ＩＣは、前記アプリケーションプログラムの一部を暗号化し、

前記コントローラは、暗号化された前記アプリケーションプログラムの一部を、前記不揮発性メモリに書き込む請求項１に記載の記憶装置。

【請求項３】前記コントローラは、前記ホスト機器からのコマンドに応じて、前記不揮発性メモリから前記暗号化されたアプリケーションプログラムの一部を読み出し、前記ＩＣへ転送し、

前記ＩＣは、前記暗号化されたアプリケーションプログラムの一部を受信し、復号化する請求項２に記載の記憶装置。

【請求項４】前記ＩＣは、中央処理装置と、前記ＩＣのオフィスシステムプログラムを格納するための第１のＲＯＭと、前記格納領域を有する電気的に書き換え可能な第２のＲＯＭと、前記アプリケーションプログラム及び前記オフィスシステムプログラムを格納するためのＲＡＭとを有する請求項１に記載の記憶装置。

【請求項５】前記セキュリティ処理は、暗号化又は復号化処理を含む請求項１に記載の記憶装置。

【請求項６】データを記憶するためのデータ領域を有する不揮発性メモリとセキュリティ処理を実行するためのＩＣと前記不揮発性メモリ及び前記ＩＣへのアクセスを制御するためのコントローラとを有し且つ着脱自在な記憶装置と、前記セキュリティ処理を前記記憶装置へ要求するためのＣＰＵと、ネットワークと通信するための通信装置とを備えた情報処理装置において、

前記不揮発性メモリは、前記アプリケーションプログラムの一部を格納するための管理領域前記ＩＣは、前記セキュリティ処理を前記ＩＣに実行させるためのアプリケーションプログラムの他部を格納するための格納領域を有する情報処理装置。

【請求項７】前記通信装置は、ネットワークを通じて、銀行取引を実行するための銀行取引サーバとクレジット決済を実行するためのクレジット決済サーバとコンテンツ配信を実行するためのコンテンツ配信サーバの少なくとも１つと通信可能である請求項６に記載の情報処理装置。

【請求項８】データ領域を有する不揮発性メモリと、前記不揮発性メモリに比較して耐タンパ性の高いＩＣと、前記不揮発性メモリ及び前記ＩＣへのアクセスを制御するためのコントローラとを備えた記憶装置において、

前記ＩＣは、銀行取引を処理するための第１のアプリケーションプログラムと、クレジット決済を処理するための第２のアプリケーションプログラムと、配信コンテンツを処理するための第３のアプリケーションプログラムとを有し、ホスト機器からのコマンドに応じて、前記第１のアプリケーションプログラム又は前記第２のアプリケーションプログラム又は前記第３のアプリケーションプログラムを選択し、実行する記憶装置。

【請求項９】前記ＩＣは、前記第１のアプリケーションプログラムを実行するための第１のＩＣと、前記第２のアプリケーションプログラムを実行するための第２のＩＣと、前記第３のアプリケーションプログラムを実行するための第３のＩＣとを有する請求項８に記載の記憶装置。

【請求項１０】前記不揮発性メモリは、前記ホスト機器からのデータを格納するためのデータ領域と、前記第１のアプリケーションプログラムの一部と前記第２のアプリケーションプログラの一部と前記第３のアプリケーションプログラムの一部との少なくとも１つを格納するための管理領域を有する請求項８に記載の記憶装置。

【請求項１１】不揮発性メモリと前記不揮発性メモリに比較して耐タンパ性の高いＩＣと前記不揮発性メモリ及び前記ＩＣへのアクセスを制御するためのコントローラと有し且つ着脱自在な記憶装置と、前記記憶装置へコマンドを発行するためのＣＰＵと、ネットワームと通信するための通信装置とを備えた情報処理装置において、

前記ＩＣは、銀行取引を処理するための第１のアプリケーションプログラムと、クレジット決済を処理するための第２のアプリケーションプログラムと、配信コンテンツを処理するための第３のアプリケーションプログラムとを有し、前記ＣＰＵからの前記コマンドに応じて、前記第１のアプリケーションプログラム又は前記第２のアプリケーションプログラム又は前記第３のアプリケーションプログラムを選択し、実行する情報処理装置。

【請求項１２】不揮発性メモリと、ＩＣと、前記不揮発性メモリ及び前記ＩＣへのアクセスを制御するためのコントローラと、前記不揮発性メモリと前記ＩＣと前記コントローラとによって共有化され且つホスト機器と接続するためのインタフェースを備えた記憶装置において、

前記コントローラは、前記ホスト機器からの第１のコマンドを受信し、前記ホスト機器からの第１のコマンドに応じて、前記ＩＣが解釈可能な第２のコマンドを作成し、前記ＩＣへ送信する記憶装置。

【請求項１３】不揮発性メモリと、処理の実行に必要なワームメモリを有するＩＣと、前記不揮発性メモリ及び前記ＩＣへのアクセスを制御するためのコントローラとを備えた記憶装置において、

前記コントローラは、前記ＩＣが解釈可能なコマンドを発行し、
前記ＩＣは、前記コマンドによって処理対象とされたデータのデータ量と前記ワームメモリの空き容量とを比較し、その比較結果に応じて前記コントローラへのレスポンスを決定し、前記コントローラへ送信する記憶装置。
【請求項１４】前記ＩＣは、前記データ量が前記空き容量より大きい場合に、前記コントローラへの処理要求を含む前記レスポンスを決定する請求項１３に記載の記憶装置。
【請求項１５】前記レスポンスは、前記ＩＣと前記コントローラとの間で予め定義されたステータスワードを含む請求項１４に記載の記憶装置。
【請求項１６】前記コントローラは、前記レスポンスを受信し、前記レスポンスによって要求された処理を実行し、その処理結果を前記ＩＣへ送信する請求項１４に記載の記憶装置。
【請求項１７】前記ＩＣは、前記データ量が前記空き容量以下の場合に、前記データに対し前記コマンドによって要求された処理を実行し、その処理結果を前記コントローラへ送信する請求項１４に記載の記憶装置。
【請求項１８】不揮発性メモリとＩＣと前記不揮発性メモリ及び前記ＩＣへのアクセスを制御するためのコントローラとを有し且つ着脱自在な記憶装置と、ＣＰＵと、ネットワームと通信するための通信装置とを備えた情報処理装置において、
前記コントローラは、前記ＩＣが解釈可能なコマンドを発行し、
前記ＩＣは、前記コマンドによって処理対象とされたデータのデータ量と前記ワームメモリの空き容量とを比較し、その比較結果に応じて前記コントローラへのレスポンスを決定し、前記コントローラへ送信する情報処理装置。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明はセキュリティ機能を搭載した記憶装置及びその記憶装置が挿入可能なホスト機器及びその記憶装置が挿入されたホスト機器に係り、不揮発性のフラッシュメモリチップ、ＩＣ（Integrated circuit;集積回路）カードチップ及びコントローラチップとを有するメモリカード及びそのメモリカードが挿入可能な（着脱自在な）情報処理装置及びそのメモリカードが挿入された情報処理装置に関する。
【０００２】
【従来の技術】ＩＣカードは、プラスチックカード基板中にＩＣ（集積回路）チップを埋め込んだものであり、その表面にＩＣチップの外部端子を持つ。ＩＣチップの外部端子には電源端子、クロック端子、データ入出力端子などがある。ＩＣチップは、接続装置が外部端子から電源や駆動クロックを直接供給することによって動作する。ＩＣカードは外部端子を通して端末機などの接続装置との間で電気信号を送受信することにより、接続装置と情報交換をおこなう。情報交換の結果として、ＩＣカードは計算結果や記憶情報の送出、記憶情報の変更をおこなう。ＩＣカードは、これらの動作仕様に基づいて、機密データ保護や個人認証などのセキュリティ処理を実行する機能を持つことができる。ＩＣカードは、クレジット決済やバンキングなど機密情報のセキュリティが必要とされるシステムにおいて、個人識別のためのユーザデバイスとして利用されている。
【０００３】
【発明が解決しようとする課題】セキュリティシステムにおいて利用されるユーザ識別装置は、秘密情報を用いて演算を行う際に、その秘密情報あるいはその秘密情報を推定できるような情報を外にもらさないように設計される必要がある。すなわち、耐タンパ性を持つことが必要とされる。また、ユーザ識別装置はユーザにとって利便性が高いことが必要である。例えば、１台の装置でより多くのセキュリティシステムに対応できるしくみを持つこと、さらに、より大きなサイズのデータを処理する能力を持つことである。
【０００４】本発明の目的は、ＩＣが実行するためのプログラムやデータ、ＩＣを管理するための情報をＩＣ外部の不揮発性メモリに保持するため、ＩＣ内の記憶容量が小さい場合にも、ＩＣが多くの処理を実行できる記憶装置及び情報処理装置を提供することである。
【０００５】本発明の目的は、ＩＣが実行する処理の一部をＩＣ外部のコントローラが実行するため、ＩＣ内の記憶容量（例えば、ＲＡＭ）が小さい場合にも、ＩＣが多くの処理を実行できる記憶装置及び情報処理装置を提供することである。
【０００６】
【課題を解決するための手段】本発明は、不揮発性メモリが、ＩＣに実行させるためのアプリケーションプログラムの一部を格納する。
【０００７】本発明は、ＩＣが、銀行取引を処理するための第１のアプリケーションプログラムと、クレジット決済を処理するための第２のアプリケーションプログラムと、配信コンテンツを処理するための第３のアプリケーションプログラムとを有し、ホスト機器からのコマンドに応じて、第１のアプリケーションプログラム又は第２のアプリケーションプログラム又は第３のアプリケーションプログラムを選択し、実行する。
【０００８】本発明は、コントローラが、ホスト機器からの第１のコマンドに応じて、ＩＣが解釈可能な第２のコマンドを作成し、ＩＣへ送信する。
【０００９】本発明は、ＩＣが、コントローラからのコマンドによって処理対象とされたデータのデータ量とＩＣ内のワームメモリの空き容量とを比較し、その比較結果に応じてコントローラへのレスポンスを決定し、コン

トローラへ送信する。

【００１０】

【発明の実施の形態】以下、本発明の一実施形態について説明する。

【００１１】図２２は、本発明を適用したＭｕｌｔｉＭｅｄｉａＣａｒｄ（ＭｕｌｔｉＭｅｄｉａＣａｒｄはＩｎｆｉｎｅｏｎＴｅｃｈｎｏｌｏｇｉｅｓＡＧの登録商標である。以下、「ＭＭＣ」と略記する。）の内部構成図を簡単に表したものである。ＭＭＣ１１０は、ＭｕｌｔｉＭｅｄｉａＣａｒｄ仕様に準拠するのが好ましい。ＭＭＣ１１０は、外部に接続したホスト機器２２０がＭｕｌｔｉＭｅｄｉａＣａｒｄ仕様に準拠したメモリカードコマンドを発行することによって、機密データ保護や個人認証などに必要な暗号演算をおこなうセキュリティ処理機能を持つ。ホスト機器２２０は、例えば、携帯電話、携帯情報端末（ＰＤＡ）、パーソナルコンピュータ、音楽再生（及び録音）装置、カメラ、ビデオカメラ、自動預金預払器、街角端末、決済端末等が該当する。ＭＭＣ１１０は、ＭＭＣ外部端子１４０、コントローラチップ１２０、フラッシュメモリチップ１３０、ＩＣカードチップ１５０を持つ。フラッシュメモリチップ１３０は、不揮発性の半導体メモリを記憶媒体とするメモリチップであり、フラッシュメモリコマンドによりデータの読み書きができる。ＭＭＣ外部端子１４０は７つの端子から構成され、外部のホスト機器２２０と情報交換するために、電源供給端子、クロック入力端子、コマンド入出力端子、データ入出力端子、グランド端子を含む。コントローラチップ１２０は、ＭＭＣ１１０内部の他の構成要素（ＭＭＣ外部端子１４０、フラッシュメモリチップ１３０、ＩＣカードチップ１５０）と接続されており、これらを制御するマイコンチップである。ＩＣカードチップ１５０は、ＩＣカードのプラスチック基板中に埋め込むためのマイコンチップであり、その外部端子、電気信号プロトコル、コマンドはＩＳＯ／ＩＥＣ７８１６規格に準拠している。ＩＣカードチップ１５０の外部端子には、電源供給端子、クロック入力端子、リセット入力端子、Ｉ／Ｏ入出力端子、グランド端子がある。コントローラチップ１２０は、ＩＣカードチップ１５０の外部端子からＩＣカードチップ１５０にＩＣカードコマンドを発行することによって、外部のホスト機器２２０から要求されたセキュリティ処理に必要な演算をおこなう。

【００１２】図２６は、本発明のＩＣカードチップの内部構成を示す図である。ＩＣカードチップ１５０は、演算処理を行うためのＣＰＵ（マイコン）１５８と、データ（プログラムを含む。）を記憶するためのＲＯＭ（Ｒｅａｄ　Ｏｎｌｙ　Ｍｅｍｏｒｙ）１５９とＲＡＭ（Ｒａｎｄｏｍ　Ａｃｃｅｓｓ　Ｍｅｍｏｒｙ）１６０とＥＥＰＲＯＭ（Ｅｌｅｃｔｒｉｃａｌｌｙ　Ｅｒａｓａｂｌｅ　Ｐｒｏｇｒａｍｍａｂｌｅ　ＲＯＭ）１６２と、暗号／復号に関する処理を行うための暗号コプロセッサ１６３と、外部とデータを送受信するためのシリアルインターフェース１６１とを備え、それらは、バス１６４によって接続される。そして、その暗号コプロセッサ１６３によって、ホスト機器２２０からのコマンドに応じて、ＩＣカードチップ１５０自らが、セキュリティ処理を実行することが可能である。尚、暗号コプロセッサ１６３（ハードウェア）の替わりに、プログラム（ソフトウェア）に従ってＣＰＵ１５８がセキュリティ処理を実行してもよい。

【００１３】一方、フラッシュメモリチップ１３０には、記憶素子を備えるが、マイコンは存在しない。

【００１４】セキュリティ処理は、例えば、ＩＣカードチップ１５０内の記憶領域にデータが書き込まれるとき、又は、ＩＣカードチップ１５０内の記憶領域からデータが読み出されるときに実行される。ＩＣカードチップ１５０のＥＥＰＲＯＭの記憶容量は、フラッシュメモリチップ１３０の記憶容量より小さい。但し、ＩＣカードチップ１５０のＥＥＰＲＯＭの記憶容量は、フラッシュメモリチップ１３０の記憶容量と同じでもよいし、大きくてもよい。

【００１５】ＩＣカードチップ１５０には、セキュリティ評価基準の国際標準であるＩＳＯ／ＩＥＣ１５４０８の評価・認証機関によって認証済みである製品を利用する。一般に、セキュリティ処理をおこなう機能を持つＩＣカードを実際の電子決済サービスなどで利用する場合、そのＩＣカードはＩＳＯ／ＩＥＣ１５４０８の評価・認証機関による評価と認定を受ける必要がある。ＭＭＣにセキュリティ処理をおこなう機能を追加することによってＭＭＣ１１０を実現し、それを実際の電子決済サービスなどで利用する場合、ＭＭＣ１１０も同様にＩＳＯ／ＩＥＣ１５４０８の評価・認証機関による評価と認定を受ける必要がある。本発明によれば、ＭＭＣ１１０は、評価・認証機関によって認証済みのＩＣカードチップ１５０を内蔵し、そのＩＣカードチップ１５０を利用してセキュリティ処理をおこなう構造を持つことにより、セキュリティ処理機能を得る。したがって、ＭＭＣ１１０はＩＳＯ／ＩＥＣ１５４０８に基づくセキュリティ評価基準を容易に満足することができ、ＭＭＣにセキュリティ処理機能を追加するための開発期間を短縮することができる。

【００１６】ＭＭＣ１１０は、ＭｕｌｔｉＭｅｄｉａＣａｒｄ仕様に準拠した外部インタフェースを持つのが好ましい。ＭＭＣ１１０は、一種類の外部インタフェースを通じて、標準メモリカードコマンド（フラッシュメモリチップ１３０へアクセスするためのコマンド）に加えて、セキュリティ処理を実行するコマンドを受け付ける必要がある。コントローラチップ１２０は、ＭＭＣ１１０が受信したコマンドが標準メモリカードコマンドであるか、セキュリティ処理を実行するコマンドであるかに

よって、アクセスすべきチップを選択し、コマンド処理を分配する機能を持つ。本発明によれば、標準メモリカードコマンドを受信したならば、フラッシュメモリチップ１３０を選択し、これにフラッシュメモリコマンドを発行してホストデータを読み書きできる。また、セキュリティ処理を実行するコマンドを受信したならば、ＩＣカードチップ１５０を選択し、これにＩＣカードコマンドを発行してセキュリティ処理を実行することができる。

【００１７】ＩＣカードチップ１５０の外部端子は、グランド端子を除いて、電源供給端子、クロック入力端子、リセット入力端子、Ｉ／Ｏ入出力端子がコントローラチップ１２０に接続されている。

【００１８】コントローラチップ１２０は、電源供給端子、クロック入力端子を通して、ＩＣカードチップ１５０への電源供給、クロック供給を制御する。本発明によれば、ホスト機器２２０からセキュリティ処理を要求されないときには、ＩＣカードチップ１５０への電源供給やクロック供給を停止させることができ、ＭＭＣ１１０の電力消費を削減することができる。

【００１９】電源供給されていないＩＣカードチップ１５０を、ＩＣカードコマンドを受信できる状態にするには、まず、ＩＣカードチップ１５０に電源供給を開始し、リセット処理を施すことが必要である。コントローラチップ１２０は、ＭＭＣ１１０がホスト機器２２０からセキュリティ処理を実行するコマンドを受信したのを契機に、電源供給端子を通してＩＣカードチップ１５０への電源供給を開始する機能を持つ。また、コントローラチップ１２０は、ＭＭＣ１１０がホスト機器２２０からセキュリティ処理を実行するコマンドを受信したのを契機に、リセット入力端子を通してＩＣカードチップ１５０のリセット処理をおこなう機能を持つ。本発明によれば、コントローラチップ１２０は、セキュリティ処理を実行するコマンドを受信するまでＩＣカードチップ１５０への電源供給を停止させておくことができる。したがって、ＭＭＣ１１０の電力消費を削減することができる。

【００２０】コントローラチップ１２０は、ＩＣカードチップ１５０のクロック入力端子を通してＩＣカードチップ１５０に供給するクロック信号をＭＭＣ１１０内部で発生し、その周波数、供給開始タイミング、供給停止タイミングを制御する機能を持つ。本発明によれば、ＭＭＣ外部端子１４０のクロック入力端子のクロック信号と無関係にすることができるため、ホスト機器２２０によるタイミング解析、電力差分解析、故障利用解析と呼ばれる攻撃法に対してセキュリティが向上する。

【００２１】図２１は、フラッシュメモリチップ１３０の詳細な内部構成を表している。フラッシュメモリチップ１３０は、ホストデータ領域２１１５と管理領域２１１０とを含む。ホストデータ領域２１１５は、セクタ単位に論理アドレスがマッピングされている領域であり、ホスト機器２２０が論理アドレスを指定してデータを読み書きできる領域である。ホストデータ領域２１１５は、ユーザファイル領域２１３０とセキュリティ処理アプリケーション領域２１２０とを含む。ユーザファイル領域２１３０は、ユーザが自由にファイルデータを読み書きできる領域である。セキュリティ処理アプリケーション領域２１２０は、ホスト機器２２０がセキュリティ処理アプリケーションに必要なデータを格納する領域であり、ユーザが不正にアクセスしないように、ホスト機器２２０のセキュリティ処理アプリケーションが論理的にユーザアクセス制限をかける。ここに格納するデータとしては、ホスト機器２２０のアプリケーションプログラム、そのアプリケーション専用のデータ、セキュリティ処理に使用される証明書など（例えば、電子決済アプリケーションプログラム、電子決済ログ情報、電子決済サービス証明書など）が可能である。本発明によれば、ＭＭＣ１１０が、ホスト機器２２０がセキュリティ処理をおこなう上で使用するデータをホスト機器２２０の代わりに格納するため、ホスト機器２２０にとって利便性が向上する。

【００２２】一方、管理領域２１１０は、コントローラチップ１２０がＩＣカードチップ１５０を管理するための情報を格納する領域である。管理領域２１１０は、ＩＣカード制御パラメータ領域２１１１、ＩＣカード環境設定情報領域２１１２、ＣＬＫ２設定情報領域２１１３、セキュリティ処理バッファ領域２１１４、セキュリティ処理ステータス領域２１１６とを含む。２１１１～２１１６の領域の詳細な使用法については後述する。

【００２３】コントローラチップ１２０は、フラッシュメモリチップ１３０の管理領域２１１０のセキュリティ処理バッファ領域２１１４を、ＩＣカードチップ１５０でセキュリティ処理を実行する際のメインメモリまたはバッファメモリとして利用する。ホスト機器２２０がセキュリティ処理を実行するコマンドによりＭＭＣ１１０にアクセスした際に、ＭＭＣ１１０がホスト機器２２０からＩＣカードチップ１５０に一度に送信できないほどの大きなサイズのセキュリティ関連データを受信したならば、コントローラチップ１２０はフラッシュメモリチップ１３０へのアクセスを選択し、そのデータを十分な容量を持つセキュリティ処理バッファ領域２１１４に一時的に格納する。ＩＣカードチップ１５０に一度に送信できないほどのサイズは、ＩＣカードコマンドの許容データサイズ（例えば、２５５バイト又は２５６バイト）を超えるサイズである。そして、コントローラチップ１２０はそれをＩＣカードチップ１５０に送信できるサイズのデータに分割し、分割データをフラッシュメモリチップ１３０から読み出し、段階的にＩＣカードチップ１５０に送信する。つまり、分割されたデータの読み出し、書き込みを繰り返す。本発明によれば、ホスト機器

２２０にとって、大きなサイズのセキュリティ関連データを扱うことができるので、セキュリティ処理の利便性が向上する。

【００２４】上記のセキュリティ処理バッファ領域２１１４を含む管理領域２１１０は、ホスト機器２２０が不正にアクセスしてセキュリティ処理を解析することができないように、コントローラチップ１２０により物理的にホストアクセス制限がかけられている。つまり、管理領域２１１０はホスト機器２２０が直接データを読み書きできない。本発明によれば、ホスト機器２２０がセキュリティ処理バッファ領域２１１４の内容を自由に読み出したり改ざんすることができないため、セキュリティ処理の信頼性や安全性が向上する。

【００２５】図２３は、ＭＭＣ１１０を利用したセキュリティ処理の一例として、コンテンツ配信のセキュリティ処理を表したものである。コンテンツプロバイダ２３１０は、ＭＭＣ１１０を所有するユーザにコンテンツ２３１４を販売する業者である。ホスト機器２２０は、この例では、コンテンツプロバイダ２３１０とネットワークなどを介して接続することができる端末機である。ユーザはＭＭＣ１１０をホスト機器２２０に接続してコンテンツ２３１４を購入する。以下、その手順を説明する。まず、ホスト機器２２０はＭＭＣ１１０に、フラッシュメモリチップ１３０に格納されたユーザ証明書２３２１を読み出すコマンドを発行する。ＭＭＣ１１０のコントローラチップ１２０は、フラッシュメモリチップ１３０のセキュリティ処理アプリケーション領域２１２０に格納されたユーザ証明書２３２１を読み出し、それをホスト機器２２０に送信する。そして、ホスト機器２２０はそれをコンテンツプロバイダ２３１０に送信する。コンテンツプロバイダ２３１０はユーザ証明書２３２１につけられたデジタル署名を検証する（２３１１）。検証が成功したならば、乱数発生器によりセッション鍵を生成し（２３１２）、それをユーザ証明書２３２１から抽出したユーザ公開鍵によって暗号化する（２３１３）。さらに、コンテンツ２３１４をそのセッション鍵によって暗号化する（２３１５）。コンテンツプロバイダ２３１０はステップ２３１３の結果をホスト機器２２０に送信する。ホスト機器２２０は、ステップ２３１３の結果をユーザ秘密鍵２３２２によって復号するセキュリティ処理を要求するコマンドを、ＭＭＣ１１０に発行する。コントローラチップ１２０は、ステップ２３１３の結果をユーザ秘密鍵２３２２によって復号するＩＣカードコマンドを、ＩＣカードチップ１５０に発行する。ＩＣカードチップ１５０は、ユーザ秘密鍵２３２２によってステップ２３１３の結果を復号して、セッション鍵を取得する（２３２３）。ホスト機器２２０は、この復号処理が成功したかを示す情報を出力させるコマンドをＭＭＣ１１０に発行する。コントローラチップ１２０は、ＩＣカードチップ１５０の出力する復号結果（復号処理が成功したかを示すＩＣカードレスポンス）をもとにしてホスト機器２２０の求める情報を構築する。そして、ＭＭＣ１１０はその情報をホスト機器２２０に送信する。次に、コンテンツプロバイダ２３１０は、ステップ２３１５の結果を、ホスト機器２２０に送信する。ホスト機器２２０は、ステップ２３１３の結果をセッション鍵（ステップ２３２３によって取得した鍵）によって復号するセキュリティ処理を要求するコマンドを、ＭＭＣ１１０に発行する。コントローラチップ１２０は、ステップ２３１５の結果をセッション鍵によって復号するＩＣカードコマンドを、ＩＣカードチップ１５０に発行する。ＩＣカードチップ１５０は、セッション鍵によってステップ２３１５の結果を復号して、コンテンツ２３１４を復元する（２３２４）。コントローラチップ１２０は、このコンテンツ２３１４をＩＣカードチップ１５０から受信し、フラッシュメモリチップ１３０に書きこむ。ホスト機器２２０は、この復号処理が成功したかを示す情報を出力させるコマンドをＭＭＣ１１０に発行する。コントローラチップ１２０は、ＩＣカードチップ１５０の出力する復号結果（復号処理が成功したかを示すＩＣカードレスポンス）をもとにしてホスト機器２２０の求める情報を構築する。そして、ＭＭＣ１１０はその情報をホスト機器２２０に送信する。ホスト機器２２０が、コンテンツを無事に受信したことをコンテンツプロバイダ２３１０に伝えると、コンテンツプロバイダ２３１０はユーザ証明書に記載されたユーザにコンテンツ料金を課金する。ユーザは、ホスト機器２２０でＭＭＣ１１０内のフラッシュメモリチップ１３０に格納されたコンテンツ２３１４を読み出して利用することができる。また、フラッシュメモリチップ１３０の記憶媒体に大容量のフラッシュメモリを使用すれば、多くのコンテンツを購入できる。本発明によれば、コンテンツ配信におけるセキュリティ処理とコンテンツ蓄積の両方をＭＭＣ１１０によって容易に実現できる。コンテンツ料金の決済を、ＩＣカードチップ１５０を利用して行ってもよい。

【００２６】図２４と図２５は、それぞれ、本発明をＳＤカード（幅２４ミリメートル、長さ３２ミリメートル、厚さ２。１ミリメートルで、９つの外部端子をもち、フラッシュメモリを搭載した小型メモリカードである。）とメモリースティック（メモリースティックはソニー株式会社の登録商標である。）に適用したときの簡単な内部構成図を表したものである。本発明を適用したＳＤカード２４１０は、ＳＤカードコントローラチップ２４２０、フラッシュメモリチップ２４３０、ＳＤカード外部端子２４４０、ＩＣカードチップ１５０とを含む。本発明を適用したメモリースティック２５１０は、メモリースティックコントローラチップ２５２０、フラッシュメモリチップ２５３０、メモリースティック外部端子２５４０、ＩＣカードチップ１５０とを含む。フラッシュメモリチップ２４３０と２５３０は、不揮発性の

半導体メモリを記憶媒体とするメモリチップであり、フラッシュメモリコマンドによりデータの読み書きができる。ＳＤカードコントローラチップ２４２０とメモリースティックコントローラチップ２５２０はそれぞれＳＤカードとメモリースティック内の他の構成要素を制御するマイコンチップである。

【００２７】ＳＤカード外部端子２４４０は９つの端子からなり、それらの位置は、端からＤａｔａ２端子２４４１、Ｄａｔａ３端子２４４２、Ｃｏｍ端子２４４３、Ｖｓｓ端子２４４４、Ｖｄｄ端子２４４５、Ｃｌｏｃｋ端子２４４６、Ｖｓｓ端子２４４７、Ｄａｔａ０端子２４４８、Ｄａｔａ１端子２４４９の順で並んでいる。Ｖｄｄ端子２４４５は電源供給端子、Ｖｓｓ端子２４４４と２４４７はグランド端子、Ｄａｔａ０端子２４４８とＤａｔａ１端子２４４９とＤａｔａ２端子２４４１とＤａｔａ３端子２４４２はデータ入出力端子、Ｃｏｍ端子２４４３はコマンド入出力端子、Ｃｌｏｃｋ端子２４４６はクロック入力端子である。ＳＤカード２４１０は、外部に接続するＳＤカードホスト機器２４６０とのインタフェース仕様にＭＭＣ１１０と違いがあるものの、ＭＭＣ外部端子１４０と非常に類似した外部端子を持ち、ＭＭＣ１１０と同様に外部からコマンドを発行することにより動作する特徴を持つため、本発明を適用することができる。

【００２８】一方、メモリースティック外部端子２５４０は１０個の端子からなり、それらの位置は、端からＧｎｄ端子２５４１、ＢＳ端子２５４２、Ｖｃｃ端子２５４３、予約端子Ｒｓｖを１つ飛ばしてＤＩＯ端子２５４４、ＩＮＳ端子２５４５、予約端子Ｒｓｖを１つ飛ばしてＳＣＫ端子２５４６、Ｖｃｃ端子２５４７、Ｇｎｄ端子２５４８の順で並んでいる。Ｖｃｃ端子２５４３と２５４７は電源供給端子、Ｇｎｄ端子２５４１と２５４８はグランド端子、ＤＩＯ端子２５４４はコマンドおよびデータ入出力端子、ＳＣＫ端子２５４６はクロック入力端子である。メモリースティック２５１０は、外部に接続するメモリースティックホスト機器２５６０とのインタフェース仕様にＭＭＣ１１０と違いがあるものの、ＭＭＣ１１０と同様に外部からコマンドを発行することにより動作する特徴を持つため、本発明を適用することができる。

【００２９】図１は、本発明を適用したＭＭＣの詳細な内部構成図を表したものである。また、図２は、図１のＭＭＣ１１０と接続したホスト機器２２０の構成とその接続状態を表したものである。ホスト機器２２０は、ＶＣＣ１電源２２１、ＣＬＫ１発振器２２２、ホストインタフェース２２３を持つ。

【００３０】ＭＭＣ１１０は、外部のホスト機器２２０と情報交換するためのＭＭＣ外部端子１４０を持つ。ＭＭＣ外部端子１４０は、ＣＳ端子１４１、ＣＭＤ端子１４２、ＧＮＤ１端子１４３および１４６、ＶＣＣ１端子１４４、ＣＬＫ１端子１４５、ＤＡＴ端子１４７の７つの端子とを含む。ＭｕｌｔｉＭｅｄｉａＣａｒｄ仕様は、ＭＭＣの動作モードとしてＭＭＣモードとＳＰＩモードという２種類を規定しており、動作モードによってＭＭＣ外部端子１４０の使用法は異なる。本実施例ではＭＭＣモードでの動作の場合について詳細に説明する。ＶＣＣ１端子１４４は、ＶＣＣ１電源２２１と接続されており、ホスト機器２２０がＭＭＣ１１０に電力を供給するための電源端子である。ＧＮＤ１端子１４３および１４６は、ＶＣＣ１電源２２１と接続されており、ＭＭＣ１１０の電気的なグランド端子である。ＧＮＤ１端子１４３とＧＮＤ１端子１４６は、ＭＭＣ１１０内部で電気的に短絡されている。ＣＳ端子１４１は、ホストインタフェース２２３に接続されており、ＳＰＩモードの動作において使用される入力端子である。ホスト機器２２０が、ＭＭＣ１１０にＳＰＩモードでアクセスするときには、ＣＳ端子１４１にＬレベルを入力する。ＭＭＣモードの動作では、ＣＳ端子１４１を使用する必要はない。ＣＭＤ端子１４２は、ホストインタフェース２２３に接続されており、ホスト機器２２０が、メモリカードインタフェース仕様に準拠したメモリカードコマンドをＭＭＣ１１０に送信したり、同仕様に準拠したメモリカードレスポンスをＭＭＣ１１０から受信するために使用する入出力端子である。ＤＡＴ端子１４７は、ホストインタフェース２２３に接続されており、ホスト機器２２０が、メモリカードインタフェース仕様に準拠した形式の入力データをＭＭＣ１１０に送信したり、同仕様に準拠した形式の出力データをＭＭＣ１１０から受信するために使用する入出力端子である。ＣＬＫ１端子１４５は、ＣＬＫ１発振器２２２に接続されており、ＣＬＫ１発振器２２２が生成するクロック信号が入力される端子である。ホスト機器２２０が、ＣＭＤ端子１４２を通してメモリカードコマンド、メモリカードレスポンスを送受信したり、ＤＡＴ端子１４７を通してホストデータを送受信するときに、ＣＬＫ１端子１４５にクロック信号が入力される。ホストインタフェース２２３には、ＣＬＫ１発振器２２２からクロック信号が供給されており、メモリカードコマンド、メモリカードレスポンス、ホストデータは、ＣＬＫ１発振器２２２が生成するクロック信号にビット単位で同期して、ホスト機器２２０とＭＭＣ１１０との間を転送される。

【００３１】ＭＭＣ１１０は、コントローラチップ１２０を持つ。コントローラチップ１２０は、ＣＰＵ１２１、フラッシュメモリＩ／Ｆ制御回路１２２、ＭＭＣＩ／Ｆ制御回路１２３、ＣＬＫ０発振器１２４、ＶＣＣ２生成器１２５、ＶＣＣ２制御回路１２６、ＣＬＫ２制御回路１２７、ＩＣカードＩ／Ｆ制御回路１２８とを含む。これらの構成要素１２１～１２８は、ホスト機器２２０からＶＣＣ１端子１４４やＧＮＤ１端子１４３、１４６を通して供給された電力により動作する。ＭＭＣＩ

／Ｆ制御回路１２３は、ＣＳ端子１４１、ＣＭＤ端子１４２、ＣＬＫ１端子１４５、ＤＡＴ端子１４７と接続されており、ＭＭＣ１１０がそれらの端子を通してホスト機器２２０と情報交換するためのインタフェースを制御する論理回路である。ＣＰＵ１２１は、ＭＭＣＩ／Ｆ制御回路１２３と接続されており、ＭＭＣＩ／Ｆ制御回路１２３を制御する。ＭＭＣＩ／Ｆ制御回路１２３がＣＭＤ端子１４２を通してホスト機器２２０からメモリカードコマンドを受信すると、ＭＭＣＩ／Ｆ制御回路１２３はそのコマンドの受信が成功したかどうかの結果をホスト機器２２０に伝えるためＣＭＤ端子１４２を通してホスト機器２２０にレスポンスを送信する。ＣＰＵ１２１は、受信したメモリカードコマンドを解釈し、コマンド内容に応じた処理を実行する。また、そのコマンド内容に応じてホスト機器２２０とＤＡＴ端子１４７を通してデータの送受信をおこなう必要がある場合、ＣＰＵ１２１は、ＭＭＣＩ／Ｆ制御回路１２３へのデータの送出、ＭＭＣＩ／Ｆ制御回路１２３からのデータの取得をおこなう。さらに、ＣＰＵ１２１は、ＭＭＣＩ／Ｆ制御回路１２３とホスト機器２２０との間のデータ転送手続きも制御する。例えば、ホスト機器２２０から受信したデータの処理中に、ホスト機器２２０がＭＭＣ１１０への電源供給を停止することがないように、ＣＰＵ１２１はＤＡＴ端子１４７にＬレベルを出力させ、ＭＭＣ１１０がビジー状態であることをホスト機器２２０に伝える。ＣＬＫ０発振器１２４は、ＣＰＵ１２１と接続され、ＣＰＵ１２１を動作させる駆動クロックを供給する。

【００３２】ＭＭＣ１１０は、フラッシュメモリチップ１３０を持つ。フラッシュメモリチップ１３０は、不揮発性の半導体メモリを記憶媒体とするメモリチップである。フラッシュメモリチップ１３０は、ホスト機器２２０からＶＣＣ１端子１４４やＧＮＤ１端子１４３、１４６を通して供給された電力により動作する。フラッシュメモリチップ１３０は、外部からのフラッシュメモリコマンドに従って、入力されたデータを不揮発性の半導体メモリに格納するライト機能、また同メモリに格納されたデータを外部に出力するリード機能を持つ。フラッシュメモリＩ／Ｆ制御回路１２２は、フラッシュメモリチップ１３０にフラッシュメモリコマンドを発行したり、そのコマンドで入出力するデータを転送するための論理回路である。ＣＰＵ１２１は、フラッシュメモリＩ／Ｆ制御回路１２２を制御し、フラッシュメモリチップ１３０にデータのライト機能やリード機能を実行させる。ホスト機器２２０から受信したデータをフラッシュメモリチップ１３０にライトしたり、フラッシュメモリチップ１３０に格納されたデータをホスト機器２２０に送信する必要があるとき、ＣＰＵ１２１は、フラッシュメモリＩ／Ｆ制御回路１２２とＭＭＣＩ／Ｆ制御回路１２３の間のデータ転送を制御する。

【００３３】ＭＭＣ１１０は、ＩＣカードチップ１５０を持つ。ＩＣカードチップ１５０は、ＩＣカードの基板中に埋め込むことを目的として設計されたマイコンチップであり、ＩＣカードの外部端子規格に準拠した８つの外部端子を持つ。このうち６つの端子は、ＩＣカードの外部端子規格により使用法が割り付けられており、残りの２つは将来のための予備端子である。その６つの端子は、ＶＣＣ２端子１５１、ＲＳＴ端子１５２、ＣＬＫ２端子１５３、ＧＮＤ２端子１５５、ＶＰＰ端子１５６、Ｉ／Ｏ端子１５７である。

【００３４】ＩＣカードチップ１５０のグランド端子は、ＭＭＣ外部端子１４０のＧＮＤ１（グランド端子）１４６に接続される。ＩＣカードチップ１５０のＶＣＣ２端子（電源入力端子）１５１は、コントローラチップ１２０のＶＣＣ２制御回路１２６に接続される。ＩＣカードチップ１５０のＲＳＴ端子（リセット入力端子）１５２とＩ／Ｏ端子（データ入出力端子）１５７は、コントローラチップ１２０のＩＣカードＩ／Ｆ制御回路１２８に接続される。ＩＣカードチップ１５０のＣＬＫ２端子（クロック入力端子）１５３は、コントローラチップ１２０のＣＬＫ２制御回路１２７に接続される。

【００３５】フラッシュメモリチップ１３０のＶＣＣ端子（電源入力端子）は、ＭＭＣ外部端子１４０のＶＣＣ１１４４に接続される。フラッシュメモリチップ１３０のＶＳＳ端子（グランド端子）は、ＭＭＣ外部端子１４０のＧＮＤ１１４６に接続される。フラッシュメモリチップ１３０のＩ／Ｏ端子（データ入出力端子）とレディ／ビジー端子とチップイネーブル端子とアウトプットイネーブル端子とライトイネーブル端子とクロック端子とリセット端子とは、コントローラチップ１２０のフラッシュメモリＩＦ制御回路１２２に接続される。

【００３６】ＶＣＣ２端子１５１は、ＩＣカードチップ１５０に電力を供給するための電源端子である。ＶＣＣ２制御回路１２６は、ＭＯＳ－ＦＥＴ素子を用いたスイッチ回路によりＶＣＣ２端子１５１への電力の供給開始と供給停止を制御する回路である。ＶＣＣ２生成器１２５はＶＣＣ２端子１５１に供給する電圧を発生し、それをＶＣＣ２制御回路１２６に供給する。ＩＣカードの電気信号規格はＩＣカードの動作クラスとしてクラスＡとクラスＢを規定している。ＶＣＣ２端子１５１に供給する標準電圧は、クラスＡでは５Ｖ、クラスＢでは３Ｖである。本発明はＩＣカードチップ１５０の動作クラスによらず適用できるが、本実施例ではＩＣカードチップ１５０がクラスＢで動作する場合について詳細に説明する。ＶＰＰ端子１５６は、ＩＣカードチップ１５０がクラスＡで動作する時に、内部の不揮発性メモリにデータを書き込んだり消去したりするために使用される可変電圧を供給する端子であり、クラスＢで動作する時には使用しない。ＧＮＤ２端子１５５は、ＩＣカードチップ１５０の電気的なグランド端子であり、ＧＮＤ１端子１４３、１４６と短絡されている。ＶＣＣ２制御回路１２６

15

はＣＰＵ１２１と接続され、ＣＰＵ１２１はＶＣＣ２端子１５１への電力供給の開始と停止を制御することができる。ＩＣカードチップ１５０を使用しないときは、ＣＰＵ１２１はＶＣＣ２端子１５１への電力供給を停止することができる。ＭＭＣ１１０は、ＩＣカードチップ１５０への電力供給を停止することにより、それが消費する電力を節約することができる。ただし、電力供給を停止すると、ＩＣカードチップ１５０の内部状態は、ＩＣカードチップ１５０内部の不揮発性メモリに記憶されたデータを除いて維持されない。

【００３７】ＣＬＫ２端子１５３は、ＩＣカードチップ１５０にクロック信号を入力する端子である。

【００３８】ＣＬＫ２制御回路１２７は、ＣＬＫ２端子１５３にクロックを供給する回路である。ＣＬＫ２制御回路１２７は、ＣＬＫ０発振器１２４から供給されたクロック信号をもとにしてＣＬＫ２端子１５３に供給するクロック信号を生成する。ＣＬＫ２制御回路１２７はＣＰＵ１２１と接続されており、ＣＬＫ２端子１５３へのクロックの供給開始と供給停止をＣＰＵ１２１から制御することができる。ＩＣカードチップ１５０は、自身内部に駆動クロック発振器をもたない。そのため、ＣＬＫ２端子１５３から駆動クロックを供給することによって動作する。ＣＬＫ２制御回路１２７が、ＣＬＫ２端子１５３へのクロック供給を停止すると、ＩＣカードチップ１５０の動作は停止するため、ＩＣカードチップ１５０の消費電力を低下させることができる。この時、ＶＣＣ２端子１５１への電力供給が保たれていれば、ＩＣカードチップ１５０の内部状態は維持される。ここで、ＣＬＫ２端子１５３に供給するクロック信号の周波数をＦ２、ＣＬＫ０発振器１２４から供給されたクロック信号の周波数をＦ０、ＰとＱを正の整数とすると、ＣＬＫ２制御回路１２７は、Ｆ２＝（Ｐ／Ｑ）＊Ｆ０の関係になるようなクロック信号を作成して、これをＣＬＫ２端子１５３に供給する。ＰとＱの値はＣＰＵ１２１により設定できるようになっている。Ｐを大きく設定してＦ２を大きくすると、ＩＣカードチップ１５０の内部処理をより高速に駆動できる。Ｑを大きく設定してＦ２を小さくすると、ＩＣカードチップ１５０の内部処理はより低速に駆動され、ＩＣカードチップ１５０の消費電力を低下させることができる。ＩＣカードチップ１５０の駆動クロック周波数は、ＩＣカードチップ１５０が正しく動作できるような許容周波数範囲内に設定される必要がある。そのため、ＣＬＫ２制御回路１２７は、Ｆ２の値がその許容周波数範囲を外れるようなＰとＱの値を設定させない特徴を持つ。

【００３９】Ｉ／Ｏ端子１５７は、ＩＣカードチップ１５０にＩＣカードコマンドを入力したり、ＩＣカードチップ１５０がＩＣカードレスポンスを出力するときに使用する入出力端子である。ＩＣカードＩ／Ｆ制御回路１２８は、Ｉ／Ｏ端子１５７と接続されており、Ｉ／Ｏ端子１５７を通してＩＣカードコマンドの信号送信やＩＣカードレスポンスの信号受信をおこなう回路である。ＩＣカードＩ／Ｆ制御回路１２８はＣＰＵ１２１に接続されており、ＣＰＵ１２１は、ＩＣカードＩ／Ｆ制御回路１２８によるＩＣカードコマンドやＩＣカードレスポンスの送受信の手続きを制御したり、送信すべきＩＣカードコマンドデータをＩＣカードＩ／Ｆ制御回路１２８に設定したり、受信したＩＣカードレスポンスをＩＣカードＩ／Ｆ制御回路１２８から取得する。ＩＣカードＩ／Ｆ制御回路１２８にはＣＬＫ２制御回路１２７からクロックが供給されており、ＩＣカードコマンドやＩＣカードレスポンスは、ＣＬＫ２端子１５３に供給するクロック信号にビット単位で同期して、Ｉ／Ｏ端子１５７を通して送受信される。また、ＲＳＴ端子１５２は、ＩＣカードチップ１５０をリセットするときにリセット信号を入力する端子である。ＩＣカードＩ／Ｆ制御回路１２８は、ＲＳＴ端子１５２と接続されており、ＣＰＵ１２１の指示によりＩＣカードチップ１５０にリセット信号を送ることができる。

【００４０】ＩＣカードチップ１５０は、ＩＣカードの電気信号規格やコマンド規格に基づいて情報交換をおこなう。ＩＣカードチップ１５０へのアクセスパターンは４種類であり、図３～図６を用いて各パターンを説明する。図３は、ＣＰＵ１２１の指示によりＩＣカードチップ１５０が非活性状態（電源が遮断されている状態）から起動して内部状態を初期化するプロセス（以下、コールドリセットと呼ぶ）において、ＩＣカードチップ１５０の外部端子の信号波形をシンプルに表したものである。図４は、ＣＰＵ１２１の指示によりＩＣカードチップ１５０が活性状態（電源が供給されている状態）で内部状態を初期化するプロセス（以下、ウォームリセットと呼ぶ）において、ＩＣカードチップ１５０の外部端子の信号波形をシンプルに表したものである。図５は、ＣＰＵ１２１の指示によりＩＣカードチップ１５０にＩＣカードコマンドを送信しＩＣカードチップ１５０からＩＣカードレスポンスを受信するプロセスにおいて、ＩＣカードチップ１５０の外部端子の信号波形をシンプルに表したものである。図６は、ＣＰＵ１２１の指示によりＩＣカードチップ１５０を非活性状態にするプロセスにおいて、ＩＣカードチップ１５０の外部端子の信号波形をシンプルに表したものである。図３～図６において、時間の方向は左から右にとっており、上の行から下の行に向かってＶＣＣ２端子１５１、ＲＳＴ端子１５２、ＣＬＫ２端子１５３、Ｉ／Ｏ端子１５７で観測される信号を表す。また、破線はそれぞれの信号の基準（Ｌレベル）を表す。

【００４１】図３を参照して、ＩＣカードチップ１５０のコールドリセット操作を説明する。まず、ＩＣカードＩ／Ｆ制御回路１２８はＲＳＴ端子１５２をＬレベルにする（３０１）。次に、ＶＣＣ２制御回路１２６はＶＣ

Ｃ２端子への電源供給を開始する（３０２）。次に、ＣＬＫ２制御回路１２７はＣＬＫ２端子１５３へのクロック信号の供給を開始する（３０３）。次に、ＩＣカードＩ／Ｆ制御回路１２８はＩ／Ｏ端子１５７を状態Ｚ（プルアップされた状態）にする（３０４）。次に、ＩＣカードＩ／Ｆ制御回路１２８はＲＳＴ端子１５２をＨレベルにする（３０５）。次に、ＩＣカードＩ／Ｆ制御回路１２８はＩ／Ｏ端子１５７から出力されるリセット応答の受信を開始する（３０６）。リセット応答の受信が終了したら、ＣＬＫ２制御回路１２７はＣＬＫ２端子１５３へのクロック信号の供給を停止する（３０７）。これで、コールドリセットの操作が完了する。なお、ステップ３０７は消費電力を低下させるための工夫であり、省略してもよい。

【００４２】図４を参照して、ＩＣカードチップ１５０のウォームリセット操作を説明する。まず、ＣＬＫ２制御回路１２７はＣＬＫ２端子１５３へのクロック信号の供給を開始する（４０１）。次に、ＩＣカードＩ／Ｆ制御回路１２８はＲＳＴ端子１５２をＬレベルにする（４０２）。次に、ＩＣカードＩ／Ｆ制御回路１２８はＩ／Ｏ端子１５７を状態Ｚにする（４０３）。次に、ＩＣカードＩ／Ｆ制御回路１２８はＲＳＴ端子１５２をＨレベルにする（４０４）。次に、ＩＣカードＩ／Ｆ制御回路１２８はＩ／Ｏ端子１５７から出力されるリセット応答の受信を開始する（４０５）。リセット応答の受信が終了したら、ＣＬＫ２制御回路１２７はＣＬＫ２端子１５３へのクロック信号の供給を停止する（４０６）。これで、ウォームリセットの操作が完了する。なお、ステップ４０６は消費電力を低下させるための工夫であり、省略してもよい。

【００４３】図５を参照して、ＩＣカードチップ１５０にＩＣカードコマンドを送信しＩＣカードチップ１５０からＩＣカードレスポンスを受信する操作を説明する。まず、ＣＬＫ２制御回路１２７はＣＬＫ２端子１５３へのクロック信号の供給を開始する（５０１）。なお、クロックがすでに供給されている場合、ステップ５０１は不要である。次に、ＩＣカードＩ／Ｆ制御回路１２８はＩ／Ｏ端子１５７にコマンドデータの送信を開始する（５０２）。コマンドデータの送信が終了したら、ＩＣカードＩ／Ｆ制御回路１２８はＩ／Ｏ端子１５７を状態Ｚにする（５０３）。次に、ＩＣカードＩ／Ｆ制御回路１２８はＩ／Ｏ端子１５７から出力されるレスポンスデータの受信を開始する（５０４）。レスポンスデータの受信が終了したら、ＣＬＫ２制御回路１２７はＣＬＫ２端子１５３へのクロック信号の供給を停止する（５０５）。これで、ＩＣカードコマンド送信とＩＣカードレスポンス受信の操作が完了する。なお、ステップ５０５は、消費電力を低下させるための工夫であり、省略してもよい。

【００４４】図６を参照して、ＩＣカードチップ１５０を非活性化する操作を説明する。まず、ＣＬＫ２制御回路１２７はＣＬＫ２端子１５３をＬレベルにする（６０１）。次に、ＩＣカードＩ／Ｆ制御回路１２８はＲＳＴ端子１５２をＬレベルにする（６０２）。次に、ＩＣカードＩ／Ｆ制御回路１２８はＩ／Ｏ端子１５７をＬレベルにする（６０３）。最後に、ＶＣＣ２制御回路１２６はＶＣＣ２端子への電源供給を停止する（６０４）。これで、非活性化の操作が完了する。

【００４５】ＩＣカードチップ１５０は、機密データ保護や個人認証などに必要な暗号演算をおこなうセキュリティ処理機能を持つ。ＩＣカードチップ１５０は、ＣＰＵ１２１との間でＩＣカードコマンドやＩＣカードレスポンスの送受信することにより情報交換をおこない、その結果として、計算の結果や記憶されている情報の送出、記憶されている情報の変更などをおこなう。ＣＰＵ１２１は、ＩＣカードチップ１５０を利用してセキュリティ処理を実行することができる。ＭＭＣ１１０がホスト機器２２０から特定のメモリカードコマンドを受信すると、ＣＰＵ１２１はそれを契機として、ＶＣＣ２制御回路１２６を通してＩＣカードチップ１５０への電源供給を制御したり、またはＣＬＫ２制御回路１２７を通してＩＣカードチップ１５０へのクロック供給を制御したり、またはＩＣカードＩ／Ｆ制御回路１２８を通してＩＣカードチップ１５０にＩＣカードコマンドを送信する。これにより、ＣＰＵ１２１は、ＩＣカードチップ１５０を利用して、ホスト機器２２０が要求するセキュリティ処理を実行する。ＣＰＵ１２１は、特定のメモリカードコマンドの受信を契機に、ＩＣカードチップ１５０に対する電源供給制御、クロック供給制御、ＩＣカードコマンド送信、ＩＣカードレスポンス受信を複数組み合わせて操作することによって、セキュリティ処理を実行してもよい。また、ＣＰＵ１２１は、ホスト機器２２０がＭＭＣ１１０へ電源供給を開始したのを契機として、セキュリティ処理を実行してもよい。セキュリティ処理の結果は、ＩＣカードチップ１５０が出力するＩＣカードレスポンスをベースにして構成され、ＭＭＣ１１０内に保持される。ＭＭＣ１１０がホスト機器２２０から特定のメモリカードコマンドを受信すると、ＣＰＵ１２１はそれを契機として、セキュリティ処理の結果をホスト機器２２０に送信する。

【００４６】図７は、ホスト機器２２０がＭＭＣ１１０にアクセスするときのフローチャートを表したものである。まず、ホスト機器２２０はＭＭＣ１１０を活性化するためにＶＣＣ１端子１４４に電源供給を開始する（７０１）。これを契機として、ＭＭＣ１１０は、第１次ＩＣカード初期化処理を実行する（７０２）。第１次ＩＣカード初期化処理の詳細は後述する。次に、ホスト機器２２０はＭＭＣ１１０を初期化するためにＣＭＤ端子１４２を通してＭＭＣ１１０の初期化コマンドを送信する（７０３）。この初期化コマンドはＭｕｌｔｉＭｅｄｉ

ａＣａｒｄ仕様に準拠したものであり、複数種類ある。ホスト機器２２０は、ＭＭＣ１１０を初期化するために、複数の初期化コマンドを送信する場合がある。ＭＭＣ１１０が初期化コマンドを受信すると、ＭＭＣ１１０はそれを処理する（７０４）。これを契機として、ＭＭＣ１１０は、第２次ＩＣカード初期化処理を実行する（７０５）。第２次ＩＣカード初期化処理の詳細は後述する。ホスト機器２２０は、ＭＭＣ１１０の初期化コマンドに対するメモリカードレスポンスを、ＣＭＤ端子１４２を通して受信し、そのメモリカードレスポンスの内容からＭＭＣ１１０の初期化が完了したかを判定する。未完了ならば、再び初期化コマンドの送信をおこなう（７０３）。ＭＭＣ１１０の初期化が完了したならば、ホスト機器２２０は、ＭｕｌｔｉＭｅｄｉａＣａｒｄ仕様に準拠した標準メモリカードコマンド（フラッシュメモリチップ１３０へアクセスするためのコマンド）や、上に述べたセキュリティ処理に関連した特定のメモリカードコマンド（ＩＣカードチップ１５０へアクセスするためのコマンド）の送信を待機する状態に移る（７０７）。この待機状態では、ホスト機器２２０は標準メモリカードコマンドを送信することができる（７０８）。ＭＭＣ１１０が標準メモリカードコマンドを受信したら、ＭＭＣ１１０はそれを処理する（７０９）。処理が完了したら、ホスト機器２２０は、再び待機状態にもどる（７０７）。この待機状態では、ホスト機器２２０はセキュリティ処理要求ライトコマンドを送信することもできる（７１０）。セキュリティ処理要求ライトコマンドとは、上に述べたセキュリティ処理に関連した特定のメモリカードコマンドの１種であり、ＭＭＣ１１０にセキュリティ処理を実行させるために処理要求を送信するメモリカードコマンドである。ＭＭＣ１１０がセキュリティ処理要求ライトコマンドを受信したら、ＣＰＵ１２１は、要求されたセキュリティ処理の内容を解釈し、セキュリティ処理をＩＣカードコマンドの形式で記述する（７１１）。即ち、ＣＰＵ１２１は、予め定められたルールに従って、ホスト機器２３０からの標準メモリカードコマンドを、ＩＣカードチップ１５０が解釈可能な特定のメモリカードコマンドへ変換する。そして、その結果として得られたＩＣカードコマンドをＩＣカードチップ１５０に発行するなどして、要求されたセキュリティ処理を実行する（７１２）。処理が完了したら、ホスト機器２２０は、再び待機状態にもどる（７０７）。この待機状態では、ホスト機器２２０はセキュリティ処理結果リードコマンドを送信することもできる（７１３）。セキュリティ処理結果リードコマンドとは、上に述べたセキュリティ処理に関連した特定のメモリカードコマンドの１種であり、ＭＭＣ１１０によるセキュリティ処理の実行結果を知るために処理結果を受信するメモリカードコマンドである。ＭＭＣ１１０がセキュリティ処理結果リードコマンドを受信したら、ＣＰＵ１２１は、ＩＣカードチップ１５０から受信したＩＣカードレスポンスをベースに、ホスト機器２２０に送信すべきセキュリティ処理結果を構築する（７１４）。そして、ホスト機器２２０は、ＭＭＣ１１０からセキュリティ処理結果を受信する。受信が完了したら、ホスト機器２２０は、再び待機状態にもどる（７０７）。なお、ステップ７１４は、ステップ７１２の中でおこなってもよい。

【００４７】図７において、ステップ７０２およびステップ７０５で実行する第１次ＩＣカード初期化処理および第２次ＩＣカード初期化処理は、ＭＭＣ１１０内でセキュリティ処理を実行するのに備えて、ＣＰＵ１２１がＩＣカードチップ１５０に対してアクセスする処理である。具体的には、ＩＣカードチップ１５０の活性化や非活性化、ＩＣカードチップ１５０のリセット、ＩＣカードチップ１５０の環境設定を行う。環境設定とは、セキュリティ処理を実行するために必要な情報（例えば、使用可能な暗号アルゴリズムの情報、暗号計算に使用する秘密鍵や公開鍵に関する情報、個人認証に使用する認証データに関する情報など）をＩＣカードチップ１５０から読み出したり、あるいはＩＣカードチップ１５０に書き込んだりすることを意味する。ＩＣカードチップ１５０の環境設定は、ＩＣカードチップ１５０にＩＣカードコマンドをＮ個（Ｎは正の整数）発行することによっておこなう。例えば、セッション鍵が３個必要ならば、ＩＣカードコマンドを３回発行し、セッション鍵が２個必要ならば、ＩＣカードコマンドを２回発行する。Ｎ個のＩＣカードコマンドは、互いに相違するものであってもよいし、同一のものであってもよい。Ｎの値は固定されたものではなく、状況によってさまざまな値となる。以下、環境設定で発行するＩＣカードコマンドを、設定コマンドと呼ぶ。また、この環境設定に基づいてセキュリティ処理を実行するＩＣカードコマンドを、以下、セキュリティコマンドと呼ぶ。セキュリティコマンドの例としては、デジタル署名の計算、デジタル署名の検証、メッセージの暗号化、暗号化メッセージの復号、パスワードによる認証などをおこなうコマンドがある。

【００４８】ＣＰＵ１２１は、ＩＣカードチップ１５０の環境設定の内容を自由に変更することができる。ＣＰＵ１２１は、セキュリティ処理の内容や結果に応じてこれを変更してもよいし、ホスト機器からのメモリカードコマンドの受信を契機としてこれを変更してもよい。また、ＣＰＵ１２１は、環境設定の内容を示した情報をフラッシュメモリチップ１３０にライトし、必要なときにフラッシュメモリチップ１３０からその情報をリードして使用することもできる。この情報は、図２１においてＩＣカード環境設定情報２１１２として示されている。これにより、ＭＭＣ１１０が非活性化されてもその情報を保持することができ、ＭＭＣ１１０が活性化されるたびにあらためて設定する手間を省くことができる。

【００４９】第１次ＩＣカード初期化処理および第２次

ＩＣカード初期化処理は、ＩＣカード制御パラメータＡ、Ｂ、Ｃに設定された値に基づいておこなわれる。また、ＣＰＵ１２１は、ステップ７１２で実行するセキュリティ処理において、ＩＣカード制御パラメータＤに設定された値に基づいてＩＣカードチップ１５０の活性化や非活性化を制御する。図８は、ＩＣカード制御パラメータの種類と設定値、それに対応した処理の内容を表している。まず、パラメータＡは、ＭＭＣ１１０に電源が供給されたときに実行される第１次ＩＣカード初期化処理に関するパラメータである。Ａ＝０のときは、ＣＰＵ１２１はＩＣカードチップ１５０にアクセスしない。Ａ＝１のときは、ＣＰＵ１２１はＩＣカードチップ１５０をコールドリセットする。Ａ＝２のときは、ＣＰＵ１２１はＩＣカードチップ１５０をコールドリセットした後でＩＣカードチップ１５０の環境設定をおこなう。Ａ＝３のときは、ＣＰＵ１２１はＩＣカードチップ１５０をコールドリセットした後でＩＣカードチップ１５０の環境設定をおこない、最後にＩＣカードチップ１５０を非活性化する。Ａ＝０またはＡ＝３のときは、第１次ＩＣカード初期化処理のあとＩＣカードチップ１５０が非活性状態となる。Ａ＝１またはＡ＝２のときは、第１次ＩＣカード初期化処理のあとＩＣカードチップ１５０は活性状態となる。次に、パラメータＢとＣは、ＭＭＣ１１０がＭＭＣ初期化コマンドを処理したときに実行される第２次ＩＣカード初期化処理に関するパラメータである。Ｂ＝０のときは、ＣＰＵ１２１はＩＣカードチップ１５０にアクセスしない。Ｂ＝１かつＣ＝１のときは、ＣＰＵ１２１はＩＣカードチップ１５０をリセット（コールドリセットまたはウォームリセット）する。Ｂ＝１かつＣ＝２のときは、ＣＰＵ１２１はＩＣカードチップ１５０をリセットした後でＩＣカードチップ１５０の環境設定をおこなう。Ｂ＝１かつＣ＝３のときは、ＣＰＵ１２１はＩＣカードチップ１５０をリセットした後でＩＣカードチップ１５０の環境設定をおこない、最後にＩＣカードチップ１５０を非活性化する。Ｂ＝２かつＣ＝２のときは、ＣＰＵ１２１はＩＣカードチップ１５０の環境設定をおこなう。Ｂ＝２かつＣ＝３のときは、ＣＰＵ１２１はＩＣカードチップ１５０の環境設定をおこなった後にＩＣカードチップ１５０を非活性化する。Ｂ＝３のときは、ＩＣカードチップ１５０が活性状態ならば、ＣＰＵ１２１はＩＣカードチップ１５０を非活性化する。最後に、パラメータＤは、ホスト機器２２０から要求されたセキュリティ処理を実行したあとに、ＩＣカードチップ１５０を非活性化するか否かを示すパラメータである。Ｄ＝０のときは、セキュリティ処理の実行後に、ＣＰＵ１２１はＩＣカードチップ１５０を非活性化せず、活性状態に保つ。Ｄ＝１のときは、セキュリティ処理の実行後に、ＣＰＵ１２１はＩＣカードチップ１５０を非活性化する。

【００５０】ＣＰＵ１２１は、ＩＣカード制御パラメータＡ、Ｂ、Ｃ、Ｄの設定値を変更することができる。ＣＰＵ１２１は、セキュリティ処理の内容や結果に応じてこれらの設定値を変更してもよいし、ホスト機器からのメモリカードコマンドの受信を契機としてこれらの設定値を変更してもよい。また、ＣＰＵ１２１は、これらの設定値をフラッシュメモリチップ１３０にライトし、必要なときにフラッシュメモリチップ１３０からこれらの設定値をリードして使用することもできる。これらの設定値は、図２１においてＩＣカード制御パラメータ２１１として示されている。これにより、ＭＭＣ１１０が非活性化されてもこれらの設定値を保持することができ、ＭＭＣ１１０が活性化されるたびにあらためて設定する手間を省くことができる。

【００５１】図９は、第１次ＩＣカード初期化処理のフローチャートを表している。初期化処理を開始する（９０１）と、まず、ＩＣカード制御パラメータＡが０かチェックする（９０２）。Ａ＝０ならばそのまま初期化処理は終了する（９０８）。Ａ＝０でないならばＩＣカードチップ１５０をコールドリセットする（９０３）。次に、ＩＣカード制御パラメータＡが１かチェックする（９０４）。Ａ＝１ならば初期化処理は終了する（９０８）。Ａ＝１でないならばＩＣカードチップ１５０の環境設定をおこなう（９０５）。次に、ＩＣカード制御パラメータＡが２かチェックする（９０６）。Ａ＝２ならば初期化処理は終了する（９０８）。Ａ＝２でないならばＩＣカードチップ１５０を非活性化する（９０７）。そして、初期化処理は終了する（９０８）。

【００５２】図１０は、第２次ＩＣカード初期化処理のフローチャートを表している。初期化処理を開始する（１００１）と、まず、ＩＣカード制御パラメータＢが０かチェックする（１００２）。Ｂ＝０ならばそのまま初期化処理は終了する（１０１３）。Ｂ＝０でないならばＢ＝１かチェックする（１００３）。Ｂ＝１ならばＩＣカード制御パラメータＡが０または３かチェックする（１００４）。Ａが０または３ならば、ＩＣカードチップ１５０をコールドリセットし（１００５）、ステップ１００７に移る。Ａが１または２ならば、ＩＣカードチップ１５０をウォームリセットし（１００６）、ステップ１００７に移る。ステップ１００７では、ＩＣカード制御パラメータＣが１かチェックする。Ｃ＝１ならば初期化処理は終了する（１０１３）。Ｃ＝１でないならばステップ１００９に移る。ステップ１００３においてＢ＝１でないならば、Ｂが２かチェックする（１００８）。Ｂ＝２ならばステップ１００９に移る。Ｂ＝２でないならば、ＩＣカード制御パラメータＡが０または３かチェックする（１０１１）。Ａが０または３ならば初期化処理を終了する（１０１３）。Ａが１または２ならば、ステップ１０１２に移る。ステップ１００９ではＩＣカードチップ１５０の環境設定をおこなう。そして、ＩＣカード制御パラメータＣが２かチェックする（１０

１０）。Ｃ＝２ならば初期化処理を終了する（１０１３）。Ｃ＝２でないならばステップ１０１２に移る。ステップ１０１２ではＩＣカードチップ１５０を非活性化する。そして、初期化処理を終了する（１０１３）。

【００５３】図１１は、ＩＣカードチップ１５０が非活性状態であるときに第１次ＩＣカード初期化処理あるいは第２次ＩＣカード初期化処理を実行した場合において、ＩＣカードチップ１５０の外部端子の信号波形をシンプルに表したものである。図１２は、ＩＣカードチップ１５０が活性状態であるときに第２次ＩＣカード初期化処理を実行した場合において、ＩＣカードチップ１５０の外部端子の信号波形をシンプルに表したものである。図１１と図１２において、時間の方向は左から右にとっており、上の行から下の行に向かってＶＣＣ２端子１５１、ＲＳＴ端子１５２、ＣＬＫ２端子１５３、Ｉ／Ｏ端子１５７で観測される信号を表す。また、横方向の破線はそれぞれの信号の基準（Ｌレベル）を表す。図１１において１１０２は図３に示したコールドリセットの信号波形を表す。図１２において１２０２は図４に示したウォームリセットの信号波形を表す。図１１と図１２において、第１設定コマンド処理１１０４ａと１２０４ａ、第２設定コマンド処理１１０４ｂと１２０４ｂ、第Ｎ設定コマンド処理１１０４ｃと１２０４ｃは、それぞれ図５に示したＩＣカードコマンド処理の信号波形を表す。ＩＣカードチップ１５０の環境設定の信号波形１１０４と１２０４は、Ｎ個の設定コマンド処理の信号波形が連なって構成される。図１１と図１２において、１１０６と１２０６は、それぞれ図６に示した非活性化の信号波形を表す。図１１と図１２において、縦方向の破線１１０１、１１０３、１１０５、１１０７、１２０１、１２０３、１２０５、１２０７はそれぞれ特定の時刻を表す。１１０１はコールドリセット前の時刻、１２０１はウォームリセット前の時刻、１１０３はコールドリセット後から環境設定前の間にある時刻、１２０３はウォームリセット後から環境設定前の間にある時刻、１１０５と１２０５は環境設定後から非活性化前の間にある時刻、１１０７と１２０７は非活性化後の時刻である。

【００５４】図１１を参照して、第１次ＩＣカード初期化処理実行時の信号波形を示す。ＩＣカード制御パラメータＡが０のときは、信号波形に変化はない。Ａ＝１のときは、時刻１１０１から時刻１１０３までの範囲の信号波形となる。Ａ＝２のときは、時刻１１０１から時刻１１０５までの範囲の信号波形となる。Ａ＝３のときは、時刻１１０１から時刻１１０７までの範囲の信号波形となる。

【００５５】図１１を参照して、ＩＣカード制御パラメータＡが０または３のときの、第２次ＩＣカード初期化処理実行時の信号波形を示す。ＩＣカード制御パラメータＢが０のときは、信号波形に変化はない。Ｂ＝１かつＩＣカード制御パラメータＣ＝１のときは、時刻１１０１から時刻１１０３までの範囲の信号波形となる。Ｂ＝１かつＣ＝２のときは、時刻１１０１から時刻１１０５までの範囲の信号波形となる。Ｂ＝１かつＣ＝３のときは、時刻１１０１から時刻１１０７までの範囲の信号波形となる。

【００５６】図１２を参照して、ＩＣカード制御パラメータＡが１または２のときの、第２次ＩＣカード初期化処理実行時の信号波形を示す。ＩＣカード制御パラメータＢが０のときは、信号波形に変化はない。Ｂ＝１かつＩＣカード制御パラメータＣ＝１のときは、時刻１２０１から時刻１２０３までの範囲の信号波形となる。Ｂ＝１かつＣ＝２のときは、時刻１２０１から時刻１２０５までの範囲の信号波形となる。Ｂ＝１かつＣ＝３のときは、時刻１２０１から時刻１２０７までの範囲の信号波形となる。Ｂ＝２かつＣ＝２のときは、時刻１２０３から時刻１２０５までの範囲の信号波形となる。Ｂ＝２かつＣ＝３のときは、時刻１２０３から時刻１２０７までの範囲の信号波形となる。Ｂ＝３のときは、時刻１２０５から時刻１２０７までの範囲の信号波形となる。

【００５７】図１３は、図７のステップ７１２において、ＣＰＵ１２１が、ホスト機器２２０が要求したセキュリティ処理をＩＣカードチップ１５０によって実行するときのフローチャートを表している。セキュリティ処理を開始する（１３０１）と、まずＩＣカードチップ１５０が非活性状態かをチェックする（１３０２）。非活性状態ならば、ＩＣカードチップ１５０をコールドリセットし（１３０３）、ステップ１３０６に移る。活性状態ならば、ステップ１３０４に移る。ステップ１３０４では、ＩＣカードチップ１５０にＩＣカードコマンドを発行する前にＩＣカードチップ１５０を再リセットする必要があるかをチェックする。必要があるならば、ＩＣカードチップ１５０をウォームリセットし（１３０５）、ステップ１３０６に移る。必要がないならば、ステップ１３０６に移る。ステップ１３０６では、ＩＣカードチップ１５０の環境設定をおこなう必要があるかをチェックする。必要があるならば、ＩＣカードチップ１５０の環境設定をおこない（１３０７）、ステップ１３０８に移る。必要がないならば、ステップ１３０８に移る。ステップ１３０８では、ＩＣカードチップ１５０のＣＬＫ２端子に供給するクロック信号の周波数Ｆ２を設定する。そして、ＣＰＵ１２１はＩＣカードチップ１５０にセキュリティコマンドを発行し、ＩＣカードチップ１５０はそれを処理する（１３０９）。セキュリティコマンドの処理時間は、クロック周波数Ｆ２に依存する。次に、ＩＣカードチップ１５０が出力するＩＣカードレスポンスにより、その処理が成功したかどうかを判定する（１３１０）。成功ならば、ステップ１３１１に移る。失敗ならば、ステップ１３１２に移る。ステップ１３１１では、ＩＣカードチップ１５０に発行すべきセキュリティコマンドが全て完了したかをチェックする。発

行すべきセキュリティコマンドがまだあるならば、ステップ１３０４に移る。発行すべきセキュリティコマンドが全て完了したならば、ステップ１３１４に移る。ステップ１３１２では、失敗したセキュリティコマンドをリトライすることが可能かを判定する。リトライできるなら、リトライ設定をおこない（１３１３）、ステップ１３０４に移る。リトライ設定とは、リトライすべきセキュリティコマンドやその関連データをＣＰＵ１２１が再度準備することである。リトライできないならステップ１３１４に移る。これは、ホスト機器２２０が要求したセキュリティ処理が失敗したことを意味する。ステップ１３１４では、ＩＣカード制御パラメータＤをチェックする。Ｄ＝１ならば、ＩＣカードチップ１５０を非活性化して（１３１５）、セキュリティ処理を終了する（１３１６）。Ｄ＝１でないならば、ＩＣカードチップ１５０を活性状態に保ったままセキュリティ処理を終了する（１３１６）。図１３のフローチャートにおいては、クロック周波数Ｆ２を、ステップ１３０９で発行するセキュリティコマンドの種類によって変えることができるように、ステップ１３０８をステップ１３０９の直前に位置させたが、ステップ１３０８はそれ以外の位置にあってもよい。

【００５８】従来のＩＣカードへの攻撃法を有効にしている要因のひとつとして、ＩＣカードの駆動クロックが外部の接続装置から直接供給されることがあげられる。駆動クロックが接続装置の制御下にあるため、タイミング解析や電力差分解析においては、電気信号の測定においてＩＣカード内部処理のタイミングの獲得が容易になる。一方、故障利用解析においては、異常な駆動クロックの供給による演算エラーの発生が容易になる。これに対し、本発明によれば、ＭＭＣ１１０内部でＩＣカードチップ１５０によりセキュリティ処理を実行するとき、ホスト機器２２０はＩＣカードチップ１５０の駆動クロックを直接供給できない。ＣＰＵ１２１は、ＩＣカードチップ１５０へ供給するクロックの周波数Ｆ２を自由に設定することができる。これにより、ホスト機器２２０の要求する処理性能に柔軟に対応したセキュリティ処理が実現できる。ホスト機器２２０が高速なセキュリティ処理を要求するならば周波数Ｆ２を高く設定し、低い消費電力を要求するならば周波数Ｆ２を低く設定したり、クロックを適度に停止させればよい。また、ＣＰＵ１２１は、周波数Ｆ２だけでなくクロックの供給開始タイミング、供給停止タイミングを自由に設定できる。これらをランダムに変化させることにより、ＩＣカードチップ１５０に対するタイミング解析、電力差分解析、故障利用解析と呼ばれる攻撃法を困難にすることができる。タイミング解析は、攻撃者が暗号処理１回の処理時間を正確に計測可能であることを仮定しているため、その対策としては、攻撃者が処理時間計測を正確に行えないようにすることが有効である。本発明によりタイミング解析が困難になる理由は、ＩＣカードチップ１５０がＩＣカードコマンドを処理している時間の長さをホスト機器２２０が正確に計測できないためである。電力差分解析の対策としては、処理の実行タイミングや順序に関する情報を外部から検出不可能にすることが有効である。本発明により電力差分解析が困難になる理由は、ＩＣカードコマンドが発行された時刻、発行されたＩＣカードコマンドの内容、発行されたＩＣカードコマンドの順序（ＩＣカードコマンドを複数組み合わせてセキュリティ処理を実行する場合）の検出がホスト機器２２０にとって困難になるためである。故障利用解析の対策としては、ＩＣカードにクロックや電圧や温度等の動作環境検知回路を搭載し、異常を検出したならば処理を停止あるいは使用不能にするという方法が有効である。本発明により故障利用解析が困難になる理由は、ＣＬＫ２制御回路１２７がＩＣカードチップ１５０に異常な駆動クロックを供給しないことが、ホスト機器２２０がＩＣカードチップ１５０に演算エラーを発生させるのを防止するからである。

【００５９】ＣＰＵ１２１は、ＩＣカードチップ１５０に供給するクロックの周波数Ｆ２、供給開始タイミング、供給停止タイミングの設定値を、セキュリティ処理の内容や結果に応じて変更してもよいし、ホスト機器からのメモリカードコマンドの受信を契機として変更してもよい。また、ＣＰＵ１２１は、これらの設定値をフラッシュメモリチップ１３０にライトし、必要なときにフラッシュメモリチップ１３０からこれらの設定値をリードして使用することもできる。これらの設定値は、図２１においてＣＬＫ２設定情報２１１３として示されている。これにより、ＭＭＣ１１０が非活性化されてもこれらの設定値を保持することができ、ＭＭＣ１１０が活性化されるたびにあらためて設定する手間を省くことができる。

【００６０】図１４は、ホスト機器２２０がセキュリティ処理要求ライトコマンドをＭＭＣ１１０に発行してから、ＩＣカードチップ１５０でセキュリティ処理が実行されるまでの過程（図７のステップ７１０～７１２）において、ＭＭＣ１１０およびＩＣカードチップ１５０の外部端子の信号波形、ＣＰＵ１２１によるフラッシュメモリチップ１３０へのアクセスをシンプルに表したものである。図１４において、時間の方向は左から右にとる。一番上の行はフラッシュメモリチップ１３０へのアクセス内容である。上から二行目の行から下の行に向かって、ＶＣＣ１端子１４４、ＣＭＤ端子１４２、ＣＬＫ１端子１４５、ＤＡＴ端子１４７、ＶＣＣ２端子１５１、ＲＳＴ端子１５２、ＣＬＫ２端子１５３、Ｉ／Ｏ端子１５７で観測される信号を表す。また、横方向の破線はそれぞれの信号の基準（Ｌレベル）を表す。図１４を参照して、ホスト機器２２０がセキュリティ処理要求ライトコマンドをＭＭＣ１１０に発行してから、ＩＣカー

ドチップ１５０でセキュリティ処理が実行されるまでの過程を説明する。まず、ホスト機器２２０はＣＭＤ端子１４２にセキュリティ処理要求ライトコマンドを送信する（１４０１）。次に、ホスト機器２２０はＣＭＤ端子１４２からセキュリティ処理要求ライトコマンドのレスポンスを受信する（１４０２）。このレスポンスは、ＭＭＣ１１０がコマンドを受信したことをホスト機器２２０に伝えるものであり、セキュリティ処理の実行結果ではない。次に、ホスト機器２２０はＤＡＴ端子１４７にセキュリティ処理要求を送信する（１４０３）。セキュリティ処理要求とは、セキュリティ処理の内容や処理すべきデータを含むホストデータである。次に、ＭＭＣ１１０はＤＡＴ端子１４７をＬレベルにセットする（１４０４）。ＭＭＣ１１０は、これによりビジー状態であることをホスト機器２２０に示す。次に、ＣＰＵ１２１は、ホスト機器２２０から受信したセキュリティ処理要求をフラッシュメモリチップ１３０にライトするコマンドを発行する（１４０５）。セキュリティ処理要求をフラッシュメモリチップ１３０にライトすることにより、ＣＰＵ１２１がセキュリティ処理要求をＩＣカードコマンド形式で記述する処理（図７のステップ７１１）において、ＣＰＵ１２１内部のワークメモリの消費量を節約できる。これは、セキュリティ処理要求のデータサイズが大きいときに有効である。なお、フラッシュメモリチップ１３０にライトされたセキュリティ処理要求は、図２１においてセキュリティ処理バッファ領域２１１４に格納される。また、ライトコマンド発行１４０５は必須な操作ではない。ライト処理期間１４０６は、フラッシュメモリチップ１３０がセキュリティ処理要求のライト処理を実行している期間を表す。セキュリティ処理１４０７はＩＣカードチップ１５０によるセキュリティ処理の信号波形を表す。この信号波形は図１３のフローチャートの遷移過程に依存する。セキュリティ処理１４０７は、ライト処理期間１４０６とオーバラップさせることができる。一般にフラッシュメモリチップ１３０のライト処理期間１４０６はミリ秒のオーダーであるため、セキュリティ処理１４０７とオーバラップさせることは、セキュリティ処理の全体的な処理時間の短縮にとって有効である。リード／ライト１４０８は、セキュリティ処理１４０７の実行中に、フラッシュメモリチップ１３０からセキュリティ処理要求をリードしたり、ＩＣカードチップ１５０が出力した計算結果をフラッシュメモリチップ１３０にライトするアクセスを示している。このアクセスにより、ＣＰＵ１２１内部のワークメモリの消費量を節約できる。これは、セキュリティ処理要求やセキュリティ処理結果のデータサイズが大きいときに有効である。リード／ライト１４０８は必須ではない。セキュリティ処理１４０７が完了したら、ＭＭＣ１１０はＤＡＴ端子１４７をＨレベルにセットする（１４０９）。ＭＭＣ１１０は、これによりセキュリティ処理が完了したことをホスト機器２２０に示す。

【００６１】図１５は、図１４におけるセキュリティ処理１４０７の信号波形の一例を表したものである。図１５において、時間の方向は左から右にとる。一番上の行はフラッシュメモリチップ１３０へのアクセス内容である。上から二行目の行から下の行に向かって、ＶＣＣ２端子１５１、ＲＳＴ端子１５２、ＣＬＫ２端子１５３、Ｉ／Ｏ端子１５７で観測される信号を表す。また、横方向の破線はそれぞれの信号の基準（Ｌレベル）を表す。１５０１は図３に示したコールドリセットの信号波形を表し、１５０４は図４に示したウォームリセットの信号波形を表し、１５０２および１５０５は図１１（あるいは図１２）に示した環境設定の信号波形を表し、１５０３および１５０６および１５０７は図５に示したＩＣカードコマンド処理の信号波形を表し、１５０８は図６に示した非活性化の信号波形を表す。ＩＣカードチップ１５０の外部端子において図１５に示した信号波形が観測されるのは、図１３のフローチャートが１３０１、１３０２、１３０３、１３０６、１３０７、１３０８、１３０９、１３１０、１３１１、１３０４、１３０５、１３０６、１３０７、１３０８、１３０９、１３１０、１３１１、１３０４、１３０６、１３０８、１３０９、１３１０、１３１１、１３１４、１３１５、１３１６の順で遷移するときである。図１５を参照して、図１４のセキュリティ処理１４０７の実行中におけるＣＰＵ１２１によるフラッシュメモリチップ１３０へのアクセス（リード／ライト１４０８）を説明する。このアクセスには、図２１におけるセキュリティ処理バッファ領域２１１４を使用する。リード１５０９、１５１１、１５１２は、それぞれ、セキュリティコマンド処理１５０３、１５０６、１５０７においてＩＣカードチップ１５０に送信するＩＣカードコマンドを構築するために必要なデータを、フラッシュメモリチップ１３０からリードするアクセスである。ライト１５１０は、セキュリティコマンド処理１５０３においてＩＣカードチップ１５０が出力した計算結果を、フラッシュメモリチップ１３０にライトするアクセスである。ライト１５１３は、セキュリティコマンド処理１５０６および１５０７においてＩＣカードチップ１５０が出力した計算結果を、フラッシュメモリチップ１３０にまとめてライトするアクセスである。リード１５０９、１５１１、１５１２は、それぞれ、セキュリティコマンド処理１５０３、１５０６、１５０７以前のＩＣカードチップ１５０へのアクセスとオーバラップさせることができる。ライト１５１０、１５１３は、それぞれ、セキュリティコマンド処理１５０３、１５０７以後のＩＣカードチップ１５０へのアクセスとオーバラップさせることができる。これらのオーバラップは、セキュリティ処理の全体的な処理時間の短縮にとって有効である。さらに、フラッシュメモリチップ１３０のライト単位が大きい場合は、ライト１５１３のように

29

複数の計算結果をまとめてライトすることができる。これは、フラッシュメモリチップ１３０へのライト回数を削減し、フラッシュメモリチップ１３０の劣化を遅らせる効果がある。なお、ライト１５１０、１５１３でフラッシュメモリチップ１３０にライトする内容は、ＩＣカードチップ１５０が出力した計算結果そのものに限定されず、図７のステップ７１５でホスト機器２２０に返すセキュリティ処理結果またはその一部であってもよい。この場合、図７のステップ７１４またはその一部は、ステップ７１２の中で実行されることになる。

【００６２】図１６は、ホスト機器２２０がセキュリティ処理結果リードコマンドをＭＭＣ１１０に発行してから、ＭＭＣ１１０がセキュリティ処理結果を出力するまでの過程（図７のステップ７１３～７１５）において、ＭＭＣ１１０の外部端子の信号波形、ＣＰＵ１２１によるフラッシュメモリチップ１３０へのアクセスをシンプルに表したものである。図１６において、時間の方向は左から右にとる。一番上の行はフラッシュメモリチップ１３０へのアクセス内容である。上から二行目の行から下の行に向かって、ＶＣＣ１端子１４４、ＣＭＤ端子１４２、ＣＬＫ１端子１４５、ＤＡＴ端子１４７で観測される信号を表す。また、横方向の破線はそれぞれの信号の基準（Ｌレベル）を表す。図１６を参照して、ホスト機器２２０がセキュリティ処理結果リードコマンドをＭＭＣ１１０に発行してから、ＭＭＣ１１０がセキュリティ処理結果を出力するまでの過程を説明する。まず、ホスト機器２２０はＣＭＤ端子１４２にセキュリティ処理結果リードコマンドを送信する（１６０１）。次に、ホスト機器２２０はＣＭＤ端子１４２からセキュリティ処理結果リードコマンドのレスポンスを受信する（１６０２）。このレスポンスは、ＭＭＣ１１０がコマンドを受信したことをホスト機器２２０に伝えるものであり、セキュリティ処理結果ではない。次に、ＭＭＣ１１０はＤＡＴ端子１４７をＬレベルにセットする（１６０３）。ＭＭＣ１１０は、これによりビジー状態であることをホスト機器２２０に示す。次に、ＣＰＵ１２１は、フラッシュメモリチップ１３０のセキュリティ処理バッファ領域（図２１の２１１４）から、ＩＣカードチップ１５０が出力した計算結果をリードする（１６０４）。ＣＰＵ１２１は、これをもとにセキュリティ処理結果を構築し、ＭＭＣ１１０がＤＡＴ端子１４７にセキュリティ処理結果を出力する（１６０５）。なお、図７のステップ７１４またはその一部が、ステップ７１２の中で実行されている場合、ステップ１６０４ではフラッシュメモリチップ１３０のセキュリティ処理バッファ領域（図２１の２１１４）からセキュリティ処理結果またはその一部をリードする。なお、フラッシュメモリチップ１３０のセキュリティ処理バッファ領域（図２１の２１１４）を利用しないでセキュリティ処理結果を構築する場合、ステップ１６０４は必要ない。

30

【００６３】ＭＭＣ１１０の製造者や管理者は、セキュリティシステムのユーザにＭＭＣ１１０を提供する前やそのユーザが所有するＭＭＣ１１０に問題が発生した時に、ＭＭＣ１１０に内蔵されたＩＣカードチップ１５０に様々な初期データを書きこんだり、ＩＣカードチップ１５０のテストをおこなったりする必要がある。ＭＭＣ１１０の製造者や管理者によるこれらの操作の利便性を高めるために、ＭＭＣ１１０は、ＩＣカードチップ１５０の外部端子をＭＭＣ外部端子１４０に割りつけるインタフェース機能を持つ。これにより、図３～図６で示したようなＩＣカードチップ１５０へのアクセス信号を、ＭＭＣ外部端子１４０から直接送受信できる。このようなＭＭＣ１１０の動作モードを、ＭｕｌｔｉＭｅｄｉａＣａｒｄ仕様に準拠した動作モードと区別して、以下、インタフェース直通モードと呼ぶ。

【００６４】インタフェース直通モードについて詳細に説明する。図１７は、ＩＣカードチップ１５０の外部端子をＭＭＣ外部端子１４０に割りつけるときの対応関係の一例を表している。この例では、ＲＳＴ端子１５２をＣＳ端子１４１に割り付け、ＧＮＤ２端子１５５をＧＮＤ１端子１４３、１４６に割り付け、ＶＣＣ２端子１５１をＶＣＣ１端子１４４に割り付け、ＣＬＫ２端子１５３をＣＬＫ１端子１４５に割り付け、Ｉ／Ｏ端子１５７をＤＡＴ端子１４７に割り付ける。このとき、ＣＳ端子１４１とＣＬＫ１端子１４５は入力端子、ＤＡＴ端子１４７は入出力端子として機能する。

【００６５】ＭＭＣ１１０は、特定のメモリカードコマンドを受信すると、動作モードをインタフェース直通モードへ移したり、インタフェース直通モードからＭｕｌｔｉＭｅｄｉａＣａｒｄ仕様に準拠した動作モードに戻すことができる。以下、動作モードをインタフェース直通モードへ移すメモリカードコマンドを直通化コマンド、動作モードをインタフェース直通モードから通常の状態に戻すメモリカードコマンドを復帰コマンドと呼ぶ。図１を参照して、ＭＭＣＩ／Ｆ制御回路１２３は、ＶＣＣ２制御回路１２６、ＣＬＫ２制御回路１２７、ＩＣカードＩ／Ｆ制御回路１２８と接続されており、ＭＭＣ１１０がホスト機器２２０から直通化コマンドを受信すると、ＣＰＵ１２１の指示により図１７で示した端子割り付けをおこなう。ＭＭＣ１１０がホスト機器２２０から復帰コマンドを受信すると、ＣＰＵ１２１の指示により図１７で示した端子割り付けを解除し、ＭＭＣ１１０はＭｕｌｔｉＭｅｄｉａＣａｒｄ仕様に準拠した動作モードに戻る。

【００６６】インタフェース直通モードでは、ホスト機器２２０がＩＣカードチップ１５０に直接アクセスできるため、セキュリティの観点からインタフェース直通モードを利用できるのは限られた者だけにする必要がある。そこで、直通化コマンドの発行には、一般のユーザに知られないパスワードの送信を必要とする。正しいパ

スワードが入力されないとインタフェース直通モードは利用できない。

【００６７】図１８は、ホスト機器２２０が、ＭＭＣ１１０の動作モードをＭｕｌｔｉＭｅｄｉａＣａｒｄ仕様に準拠した動作モードからインタフェース直通モードに移し、ＩＣカードチップ１５０に直接アクセスし、その後、ＭＭＣ１１０の動作モードを再びＭｕｌｔｉＭｅｄｉａＣａｒｄ仕様に準拠した動作モードに戻すまでの処理のフローチャートを表している。ホスト機器２２０は処理を開始し（１８０１）、まずＭＭＣ１１０に直通化コマンドを発行する（１８０２）。ＭＭＣ１１０は、直通化コマンドで送信されたパスワードが正しいかチェックする（１８０３）。正しければステップ１８０４に移り、間違っていれば処理は終了する（１８１０）。ステップ１８０４では、ＣＰＵ１２１は、ＩＣカードチップ１５０をコールドリセットする。そして、図１７で示した端子割り付けをおこないインタフェースを直通化する（１８０５）。この時点から、ホスト機器２２０はＩＣカードチップ１５０に直接アクセスする（１８０６）。ホスト機器２２０がＩＣカードチップ１５０への直接アクセスを終了し、ＭＭＣ１１０の動作モードを再びＭｕｌｔｉＭｅｄｉａＣａｒｄ仕様に準拠した動作モードに戻すときは、ＭＭＣ１１０に復帰コマンドを発行する（１８０７）。すると、ＣＰＵ１２１は図１７で示した端子割り付けを解除し、ＭＭＣ１１０はＭｕｌｔｉＭｅｄｉａＣａｒｄ仕様に準拠した動作モードに戻る（１８０８）。そして、ＣＰＵ１２１は、ＩＣカードチップ１５０を非活性化する（１８０９）。以上で、処理は終了する（１８１０）。

【００６８】図１９は、図１８のステップ１８０１～１８０６の過程において、ＭＭＣ１１０およびＩＣカードチップ１５０の外部端子の信号波形をシンプルに表したものである。図１９において、時間の方向は左から右にとる。上の行から下の行に向かって、ＶＣＣ１端子１４４、ＣＭＤ端子１４２、ＣＬＫ１端子１４５、ＤＡＴ端子１４７、ＶＣＣ２端子１５１、ＲＳＴ端子１５２、ＣＬＫ２端子１５３、Ｉ／Ｏ端子１５７で観測される信号を表す。また、横方向の破線はそれぞれの信号の基準（Ｌレベル）を表す。１９０５は、図３のコールドリセットの信号波形を示す。モード移行時刻１９０６は、動作モードがインタフェース直通モードに移る時刻を表す。

【００６９】図１９を参照して、ホスト機器２２０がＭＭＣ１１０の動作モードをＭｕｌｔｉＭｅｄｉａＣａｒｄ仕様に準拠した動作モードからインタフェース直通モードに移しＩＣカードチップ１５０に直接アクセスする過程を説明する。なお、ＭＭＣ１１０のＶＣＣ１端子１４４には３Ｖ（ＶＣＣ２端子１５１の標準電圧）が供給されている。ホスト機器２２０がＣＭＤ端子１４２に直通化コマンドを入力すると（１９０１）、ＣＭＤ端子１４２から直通化コマンドのレスポンスが出力される（１９０２）。このレスポンスは、ＭＭＣ１１０がコマンドを受信したことをホスト機器２２０に伝えるものである。次に、ホスト機器２２０はＤＡＴ端子１４７にパスワードを入力する（１９０３）。パスワード入力後、ＭＭＣ１１０はＤＡＴ端子１４７にＬレベルを出力し（１９０４）、ビジー状態であることをホスト機器２２０に示す。ビジー状態の間に、ＣＰＵ１２１は、ＩＣカードチップ１５０をコールドリセットする（１９０５）。そして、モード移行時刻１９０６において、動作モードをインタフェース直通モードに移す。このときに、ＤＡＴ端子１４７はＬレベルからハイインピーダンス状態になる。これにより、ホスト機器２２０はビジー状態の解除を知ることができる。この時点から、ホスト機器２２０はＩＣカードチップ１５０に直接アクセスする。例えば、ＣＬＫ１端子１４５にクロックを供給すると（１９０７）、ＣＬＫ２端子１５３にそのクロックが供給される（１９０８）。また、ＤＡＴ端子１４７にＩＣカードコマンドを送信すると（１９０９）、Ｉ／Ｏ端子１５７にそのＩＣカードコマンドが送信される（１９１０）。

【００７０】図２０は、図１８のステップ１８０７～１８１０の過程において、ＭＭＣ１１０およびＩＣカードチップ１５０の外部端子の信号波形をシンプルに表したものである。図２０において、時間の方向は左から右にとる。上の行から下の行に向かって、ＶＣＣ１端子１４４、ＣＭＤ端子１４２、ＣＬＫ１端子１４５、ＤＡＴ端子１４７、ＶＣＣ２端子１５１、ＲＳＴ端子１５２、ＣＬＫ２端子１５３、Ｉ／Ｏ端子１５７で観測される信号を表す。また、横方向の破線はそれぞれの信号の基準（Ｌレベル）を表す。モード復帰時刻２００３は、動作モードがインタフェース直通モードからＭｕｌｔｉＭｅｄｉａＣａｒｄ仕様に準拠した動作モードに戻る時刻を表す。２００４は、図６の非活性化の信号波形を示す。

【００７１】図２０を参照して、ホスト機器２２０がＭＭＣ１１０の動作モードをインタフェース直通モードからＭｕｌｔｉＭｅｄｉａＣａｒｄ仕様に準拠した動作モードに戻す過程を説明する。なお、ＭＭＣ１１０のＶＣＣ１端子１４４には３Ｖ（ＶＣＣ２端子１５１の標準電圧）が供給されている。ホスト機器２２０がＣＭＤ端子１４２に復帰コマンドを入力すると（２００１）、ＣＭＤ端子１４２から復帰コマンドのレスポンスが出力される（２００２）。このレスポンスは、ＭＭＣ１１０がコマンドを受信したことをホスト機器２２０に伝えるものである。そして、モード復帰時刻２００３において、ＭＭＣ１１０はＤＡＴ端子１４７にＬレベルを出力してビジー状態であることをホスト機器２２０に示し、それと同時に動作モードをＭｕｌｔｉＭｅｄｉａＣａｒｄ仕様に準拠した動作モードに戻す。ビジー状態の間に、ＣＰＵ１２１は、ＩＣカードチップ１５０を非活性化する（２００４）。そして、ＭＭＣ１１０は、ＤＡＴ端子１

４７をハイインピーダンス状態にし（２００５）、復帰コマンドの処理が完了したことをホスト機器２２０に示す。これ以後、ホスト機器２２０はＩＣカードチップ１５０に直接アクセスできない。ホスト機器２２０が、ＣＬＫ１端子１４５にクロックを供給しながらＣＭＤ端子１４２に何らかのメモリカードコマンドを送信した場合、ＩＣカードチップ１５０にそのクロック信号（２００６）は伝わらない。２００１や２００２においてホスト機器２２０がＣＬＫ１端子１４５に供給するクロック信号は、ＩＣカードチップ１５０のＣＬＫ２端子１５３にも伝わるが、ＤＡＴ端子１４７がハイインピーダンス状態であるため、ＩＣカードチップ１５０がＩＣカードコマンドを誤って認識することはない。

【００７２】図２１において、セキュリティ処理ステータス領域２１１６には、ＩＣカードチップ１５０によるセキュリティ処理の進捗状況を示す情報を格納する。ＣＰＵ１２１は、この情報をセキュリティ処理の実行中に更新することができる。例えば、セキュリティ処理の途中でＭＭＣ１１０への電源供給が停止した場合、電源供給再開時にＣＰＵ１２１がこの情報をリードして参照すれば、セキュリティ処理を中断した段階から再開することができる。

【００７３】本発明の実施形態によれば、メモリカード外部からＩＣカードチップの駆動クロックを直接供給しないため、ＩＣカードチップの処理時間を正確に計測できず、また、処理の実行タイミングや順序の検出が困難になる。さらに、異常な駆動クロックを供給することができず、演算エラーを発生させるのが困難になる。したがって、タイミング解析、電力差分解析、故障利用解析攻撃法に対するセキュリティが向上する。

【００７４】本発明の実施形態によれば、メモリカード外部からＩＣカードチップの制御方式を自由に設定できる。例えば、高速処理が要求されるならば、ＩＣカードチップの駆動クロックの周波数を高くした制御方式を設定し、低消費電力が要求されるならば、ＩＣカードチップの駆動クロックの周波数を低くしたり、ＩＣカードチップの駆動クロックを適度に停止させる制御方式を設定することができる。したがって、セキュリティシステムの要求する処理性能に柔軟に対応したセキュリティ処理が実現できる。

【００７５】本発明によれば、ＩＣカードチップによるセキュリティ処理に必要なデータや、ＩＣカードチップを管理するための情報を、フラッシュメモリに保持することができる。したがって、セキュリティ処理の利便性を向上させることができる。

【００７６】本発明の実施形態によれば、ＭＭＣの製造者や管理者が、ＭＭＣ内部のＩＣカードチップに直接アクセスすることができる。したがって、ＭＭＣ内部のＩＣカードチップの初期化やメンテナンスを、従来のＩＣカードと同様な方法で実現できる。

【００７７】本発明の実施形態によれば、フラッシュメモリチップを備えたＭＭＣに、セキュリティ機能を追加する場合、セキュリティ評価機関の認証を予め受けたＩＣカードチップ追加搭載することによって、セキュリティ評価機関によるＭＭＣの認証が不要となるため、ＭＭＣの開発期間又は製造期間が短縮する。

【００７８】本発明を適用したＭＭＣ１１０は、高度な情報セキュリティが必要とされる銀行取引、クレジット決済、コンテンツ配信など多様な電子商取引サービスに適用することができる。ユーザにとっては、これらの多様なサービス毎に異なるＭＭＣ１１０を持つよりも、１枚で複数のサービスが利用できるＭＭＣ１１０を持つほうが便利である。１枚のＭＭＣ１１０を複数のサービスに利用することを考えたとき、一般にサービス毎にその事業者は異なるため、ＭＭＣ１１０に内蔵されたＩＣカードチップ１５０の内部では、複数のサービス事業者が各自のサービス用に設計開発したセキュリティ処理プログラムが共存することになる（以下、ＩＣカードチップ１５０内のセキュリティ処理プログラムをＩＣカードアプリケーションと呼ぶ）。したがって、ＩＣカードチップ１５０には、サービスを利用するユーザが複数のＩＣカードアプリケーションから使用したいものを選択できる機能（アプリケーション選択機能）や、複数のＩＣカードアプリケーションが互いに干渉して誤動作を起こしたり秘密情報が解読されたりしないようにする機能（ファイヤウォール機能）が必要となる。そこで、ＩＣカードチップ１５０内に、アプリケーション選択機能やファイヤウォール機能を持つオペレーティングシステム（以下、ＩＣカードＯＳと呼ぶ。）を設置し、各ＩＣカードアプリケーションは、その上で実行可能なプログラムとして実装する。ＩＣカードＯＳとしては、ＭＵＬＴＯＳ（ＭＵＬＴＯＳはＭｏｎｄｅｘ　ＩｎｔｅｒｎａｔｉｏｎａｌＬｉｍｉｔｅｄの登録商標である。）や、ＪａｖａＣａｒｄの仮想マシン（ＪａｖａＣａｒｄはＳｕｎ　Ｍｉｃｒｏｓｙｓｔｅｍｓ株式会社の登録商標である。）などを適用することが望ましい。

【００７９】ＩＣカードＯＳ上で実行可能なプログラムは一般にＩＣカード内のＥＥＰＲＯＭに格納されるが、本発明では、ＩＣカードアプリケーションのプログラムはＩＣカードチップ１５０内のＥＥＰＲＯＭ１６２に格納されるだけでなく、プログラムの一部がフラッシュメモリチップ１３０にも格納される。これにより、１つのＩＣカードアプリケーションが消費するＥＥＰＲＯＭ１６２の量を削減することができ、ＩＣカードチップ１５０に搭載可能なＩＣカードアプリケーションの数を増やすことができる。よって、ユーザは１枚のＭＭＣ１１０でより多くのサービスを利用できる。

【００８０】以下、図２７～図３７を参照しながら、ＩＣカードアプリケーションのプログラムの一部をフラッシュメモリチップ１３０に格納したＭＭＣ１１０のアプ

リケーション管理機能について詳しく説明する。

【００８１】図２７は、ＩＣカードチップ１５０の機能的構成を中心にして、ＭＭＣ１１０の内部構成を示したものである。ＩＣカードインタフェース１７１は、ＩＣカードチップ１５０の外部端子（ＶＣＣ２、ＲＳＴ、ＣＬＫ２、ＧＮＤ２、Ｉ／Ｏ）、シリアルインタフェース１６１、ＩＣカードコマンド／レスポンスの送受信を制御するプログラムをまとめて表したものである。ＩＣカードＯＳ１７２は、上で述べたＩＣカードＯＳを表し、ＩＣカードインタフェース１７１を通して外部から受信したＩＣカードコマンドに応じて、ＩＣカードアプリケーションの選択をおこなう。また、選択されたＩＣカードアプリケーションへコマンドデータの引き渡しをおこない、選択されたＩＣカードアプリケーションから受け取った処理結果データに基づいてＩＣカードレスポンス作成し、ＩＣカードインタフェース１７１を通して外部へ送信する。アプリケーションＡ１７３、アプリケーションＢ１７４、アプリケーションＣ１７５は、それぞれＩＣカードＯＳ１７２の上で実行可能なプログラムとして実装されたＩＣカードアプリケーションであり、ユーザが利用したいサービスに応じてＭＭＣ１１０外部からメモリカードコマンドで選択することができる（例えば、銀行取引サービスを利用したい時はアプリケーションＡ１７３を、クレジット決済サービスを利用したい時はアプリケーションＢ１７４を、コンテンツ配信サービスを利用したい時はアプリケーションＣ１７５を選択するなど）。次に、アプリケーションＡ１７３内のコマンド解釈部１８１、モジュール実行部１８２、モジュールロード部１８３、モジュール管理情報１８４、モジュールロード領域１８５、認証部１８６およびフラッシュメモリチップ１３０内のコマンド処理モジュール１８７は、アプリケーションＡ１７３のプログラムの一部をフラッシュメモリチップ１３０に格納するという機能を実現するための機能ブロックやデータを表している。フラッシュメモリチップ１３０に格納されているアプリケーションＡ１７３のプログラムの一部をコマンド処理モジュール１８７と呼ぶ。コマンド処理モジュール１８７は、アプリケーションＡ１７３が選択された時に使用することができるＩＣカードコマンド中の１つのコマンドを処理するプログラムモジュールであり、コマンド毎に用意されている。コマンド処理モジュール１８７はフラッシュメモリチップ１３０に複数格納することができる。コマンド処理モジュール１８７はフラッシュメモリチップ１３０に格納されているときは暗号化されており、コマンド処理の内容はアプリケーションＡ１７３に対応するサービスの事業者以外に知られないようになっており、アプリケーションＡ１７３内で暗号が解かれてモジュールロード領域１８５にロードされてはじめて実行が可能となる。各機能ブロックやデータがどのように作用するかについては、以下の電子取引サービスの具体例を用いて後で詳しく説明する。

【００８２】電子取引サービスの具体例として銀行取引サービスを示す。図２８～図３０を用いてそのサービスを詳細に説明する。

【００８３】図２８は、ＭＭＣとのインタフェース持つ携帯端末を利用してユーザが銀行取引サービスを実行するシステムの構成を表している。ホスト機器２２０は、ＭＭＣとのインタフェースを持つ携帯端末であり、ホストインタフェース２２３を通してＭＭＣ１１０にコマンドによってアクセスする。銀行取引サーバ２８３０は、ユーザの銀行口座へのアクセスを提供し、ユーザからの指示に応じて銀行取引を実行する。ホスト機器２２０は通信手段２８１３を持つ。通信手段２８１３は、ホスト機器２２０がネットワーク２８２０を通して銀行取引サーバ２８３０に接続して情報交換をおこなう際の通信処理をおこなう。ホスト機器２２０は、ＣＰＵ２８１１をもつ。ＣＰＵ２８１１は、ホストインタフェース２２３や通信手段２８１３を制御し、ＭＭＣ１１０や銀行取引サーバ２８３０との間で情報交換をおこなう。ホスト機器２２０は情報表示手段２８１４を持つ。情報表示手段２８１４は、ＣＰＵ２８１１により制御され、銀行取引に関する情報をユーザに表示する。ホスト機器２２０はユーザ入力手段２８１２を持つ。ユーザ入力手段２８１２から入力されたデータはＣＰＵ２８１１で処理される。ユーザは、ユーザ入力手段２８１２を用いてユーザ認証のための暗証番号を入力したり、所望の銀行取引（残高照会、振込みなど）を指示する。

【００８４】図２９および図３０は、図２８のシステムにおいて銀行取引サービスを実行するときの手順を示したフローチャートである。ここでは簡単のため、エラー発生時の処理フローを省略する。まず、ユーザは、ＭＭＣ１１０を利用したセキュリティ処理をおこなうため、第１ＰＩＮ（第１の暗証番号）によってＭＭＣ１１０から認証されなければならない。ホスト機器２２０は、ユーザがユーザ入力手段２８１２から入力した第１ＰＩＮをＭＭＣ１１０へ送信するため、第１ＰＩＮ検証コマンドを発行する（２９０１）。ＭＭＣ１１０は、内部のＩＣカードチップ１５０に第１ＰＩＮ検証のためのＩＣカードコマンドを送信し、受信した第１ＰＩＮが正しいかを検証する（２９０２）。検証が成功すると、銀行取引サービスに対応したＩＣカードアプリケーションによるセキュリティ処理の利用が許可される。そこで、ホスト機器２２０は、ＭＭＣ１１０に銀行取引アプリケーション選択コマンドを発行する（２９０３）。ＭＭＣ１１０は、内部のＩＣカードチップ１５０にアプリケーション選択のためのＩＣカードコマンドを送信し、銀行取引サービスに対応したＩＣカードアプリケーションを利用可能な状態にする（２９０４）。次に、ホスト機器２２０は、銀行取引サーバ２８３０に銀行取引開始要求のメッセージを送信する（２９０５）。銀行取引サーバ２８３

０は、認証局が発行したサーバ証明書（サーバ公開鍵を含む）をホスト機器２２０に送信する（２９０６）。ホスト機器２２０は、サーバ証明書を検証するため、サーバ証明書のハッシュ値を計算する（２９０７）。そして、ホスト機器２２０は、ＭＭＣ１１０に署名検証コマンドを発行して、そのハッシュ値とサーバ証明書につけられた認証局による署名を送信する（２９０８）。ＭＭＣ１１０は、内部のＩＣカードチップ１５０に署名検証のためのＩＣカードコマンドを送信し、受信した署名が正しいかを認証局の公開鍵を用いて検証する（２９０９）。次に、ホスト機器２２０は、ＭＭＣ１１０に乱数発生コマンドを発行する（２９１０）。ＭＭＣ１１０は、内部のＩＣカードチップ１５０に乱数発生のためのＩＣカードコマンドを送信する。ＩＣカードチップ１５０は乱数を発生し、それをステップ２９１４のために一時的に保持する（２９１１）。そして、ホスト機器２２０は、サーバ証明書からサーバ公開鍵を抽出し（２９１２）、ＭＭＣ１１０に暗号化コマンドを発行してサーバ公開鍵を渡す（２９１３）。ＭＭＣ１１０は、内部のＩＣカードチップ１５０に暗号化のためのＩＣカードコマンドを送信し、ステップ２９１１で発生した乱数をサーバ公開鍵で暗号化させ、ホスト機器２２０に返す（２９１４）。ホスト機器２２０は、暗号化した乱数と、認証局が発行したユーザ証明書（ユーザ公開鍵を含む）を銀行取引サーバ２８３０に送信する（２９１５）。銀行取引サーバ２８３０は、サーバ秘密鍵で暗号化乱数を復号して乱数を取得し、ユーザ証明書を認証局の公開鍵を用いて検証し、ユーザ証明書の検証が成功すれば、そこからユーザ公開鍵を抽出し、取得した乱数をユーザ公開鍵で暗号化する（２９１６）。そして図３０に移って、銀行取引サーバ２８３０は、ユーザ公開鍵で暗号化した乱数をホスト機器２２０に送信する（３００１）。ホスト機器２２０はこの暗号化乱数を受信する（３００２）。そして、ホスト機器２２０は、銀行取引サーバ２８３０が本物かどうかを知るため暗号化乱数から乱数を復元できるかを調べる。そこで、ＭＭＣ１１０に復号・比較コマンドを発行して、暗号化乱数を送信する（３００３）。ＭＭＣ１１０は、内部のＩＣカードチップ１５０に復号・比較のためのＩＣカードコマンドを送信する。ＩＣカードチップ１５０はユーザ秘密鍵で暗号化乱数を復号し、ステップ２９１１で発生した乱数と比較し、一致したかどうかの結果を返す（３００４）。ホスト機器２２０は、ＭＭＣ１１０から乱数が一致したことを示す応答を受け取ると、情報表示手段２８１４を用いてユーザに銀行との取引内容を指示するよう求める。ユーザはユーザ入力手段２８１２から取引内容を指示する。そして、ホスト機器２２０は、指示された取引内容（残高照会、振込みなど）を銀行取引サーバ２８３０に送信する（３００５）。銀行取引サーバ２８３０は、受信した取引内容が本当にユーザからのものかを確認するため、取引内容を記した銀行取引契約書を作成し、作成した銀行取引契約書へのユーザの署名を求めるため、作成した銀行取引契約書をホスト機器２２０に送信する（３００７）。ホスト機器２２０は、受信した銀行取引契約書を情報表示手段２８１４に表示する（３００８）。そしてユーザに内容確認および署名を求める。ユーザは、内容が正しいことを確認したならば、ＭＭＣ１１０を利用してユーザ秘密鍵による銀行取引契約書への電子的な署名をおこなう。ＭＭＣ１１０は署名処理の実行を許可する前に、第２のＰＩＮによって再度ユーザを認証する。ユーザはユーザ入力手段２８１２から第２ＰＩＮを入力する。ホスト機器２２０は、入力された第２ＰＩＮをＭＭＣ１１０へ送信するため、第２ＰＩＮ検証コマンドを発行する（３００９）。ＭＭＣ１１０は、内部のＩＣカードチップ１５０に第２ＰＩＮ検証のためのＩＣカードコマンドを送信し、受信した第２ＰＩＮが正しいかを検証する（３０１０）。検証が成功すると、ユーザ秘密鍵による署名処理の利用が許可される。ホスト機器２２０は、銀行取引契約書への署名を作成するために銀行取引契約書のハッシュ値を計算する（３０１１）。そして、ホスト機器２２０は、ＭＭＣ１１０に署名作成コマンドを発行して、そのハッシュ値を送信する（３０１２）。ＭＭＣ１１０は、内部のＩＣカードチップ１５０に署名作成のためのＩＣカードコマンドを送信し、ハッシュ値とユーザ秘密鍵により署名を作成する（３０１３）。ホスト機器２２０は、作成した署名をつけた銀行取引契約書を銀行取引サーバ２８３０に送信する（３０１４）。銀行取引サーバ２８３０は、受信した銀行取引契約書につけられた署名を、図２９のステップ２９１６で取得したユーザ公開鍵を用いて検証する（３０１５）。検証が成功すれば、その取引内容は本物のユーザが指示したものであることが証明され、銀行取引サーバ２８３０は、指示された銀行取引の処理を実行する（３０１６）。処理が完了したら、銀行取引サーバ２８３０は、完了通知のメッセージをホスト機器２２０に送信する（３０１７）。ホスト機器２２０は、完了通知のメッセージを受信して情報表示手段２８１４にそれを表示し、ユーザに取引の完了を通知する（３０１８）。以上が、銀行取引サービスの実行手順である。

【００８５】図２７において、例えば、アプリケーションＡ１７３が、以上の銀行取引サービスを実行するためのＩＣカードアプリケーションであるとする。以下、銀行取引サービスを例にして、アプリケーションＡ１７３のプログラムの一部をフラッシュメモリチップ１３０に格納するという機能について具体的に説明していく。

【００８６】図３１は、図２７のコマンド処理モジュール１８７がフラッシュメモリチップ１３０のどこに格納されるかを詳細に示したものである。アプリケーションＡ用コマンド処理モジュール群３１１１は、銀行取引サービスの実行手順においてＭＭＣ１１０内のＩＣカード

チップ１５０に送信されるＩＣカードコマンドを処理するためのコマンド処理モジュール１８７を、複数まとめたものである。図３２は、アプリケーションＡ用コマンド処理モジュール群３１１１の内容を示している。銀行取引サービスでは７つのＩＣカードコマンドが使用されるため、７つのコマンド処理モジュール１８７が含まれている。第１ＰＩＮ検証処理モジュール３２０１は、図２９のステップ２９０２で使用する。署名検証処理モジュール３２０２は、図２９のステップ２９０９で使用する。乱数発生処理モジュール３２０３は、図２９のステップ２９１１で使用する。暗号化処理モジュール３２０４は、図２９のステップ２９１４で使用する。復号・比較処理モジュール３２０５は、図３０のステップ３００４で使用する。第２ＰＩＮ検証処理モジュール３２０６は、図３０のステップ３０１０で使用する。署名作成処理モジュール３２０７は、図３０のステップ３０１３で使用する。これらのコマンド処理モジュール１８７は識別番号１から７によって管理され、アプリケーションＡ用コマンド処理モジュール群３１１１の中から必要なものを引き出すことができる。図３１において、アプリケーションＢ用コマンド処理モジュール群３１１２、アプリケーションＣ用コマンド処理モジュール群３１１３も、アプリケーションＡ用コマンド処理モジュール群３１１１と同様に、それぞれのＩＣカードアプリケーションに対応する電子取引サービスで使用されるＩＣカードコマンドを処理するためのコマンド処理モジュール１８７を、複数まとめたものである。

【００８７】コマンド処理モジュール１８７を使用するためにはＩＣカードチップ１５０へロードする必要がある。このロード処理は、図１３のステップ１３０６や１３０７で示した環境設定の一種である。そのため、アプリケーションＡ用コマンド処理モジュール群３１１１、アプリケーションＢ用コマンド処理モジュール群３１１２、アプリケーションＣ用コマンド処理モジュール群３１１３といったコマンド処理モジュール群は、図３１のように、フラッシュメモリチップ１３０の管理領域２１１０におけるＩＣカード環境設定情報領域２１１２に格納する。管理領域２１１０以外の領域（ホストデータ領域２１１５）に格納してもよいが、不正に改ざんされたりすることを防止するため、管理領域２１１０に格納するほうが望ましい。モジュールデータ領域３１１０は、各コマンド処理モジュール群を格納するために、ＩＣカード環境設定情報領域２１１２内に用意された領域である。なお、図３１では図２１に示した領域のうち、説明に必要ないものを省略している。

【００８８】次に、ＩＣカードチップ１５０に登録されたアプリケーションＡ１７３の中にあるモジュールロード領域１８５について詳細に説明する。図３３は、ＩＣカードチップ１５０内のメモリ資源（ＲＯＭ１５９、ＲＡＭ１６０、ＥＥＰＲＯＭ１６２）の詳細な内部構成を示したものである。まず、ＲＯＭ１５９には、ＩＣカードＯＳプログラム３３４１、ＩＣカードインタフェース制御プログラム３３４２が含まれる。ＩＣカードＯＳプログラム３３４１は、ＩＣカードＯＳ１７２を機能させるためにＣＰＵ１５８で実行されるプログラムである。ＩＣカードインタフェース制御プログラム３３４２は、ＩＣカードインタフェース１７１を制御するためにＣＰＵ１５８で実行されるプログラムである。次に、ＲＡＭ１６０には、ＩＣカードＯＳワーク領域３３５１、アプリケーションワーク領域３３５２が含まれる。ＩＣカードＯＳワーク領域３３５１は、ＣＰＵ１５８がＩＣカードＯＳプログラム３３４１を実行するときに使用するメインメモリである。アプリケーションワーク領域３３５２は、ＩＣカードＯＳ１７２によって選択されたＩＣカードアプリケーション（１７３、１７４、１７５など）が使用するメインメモリである。次に、ＥＥＰＲＯＭ１６２には、ＩＣカードＯＳ設定情報３３１０、アプリケーションＡ１７３、アプリケーションＢ１７４、アプリケーションＣ１７５などが含まれる。ＩＣカードＯＳ設定情報３３１０は、ＩＣカードＯＳ１７２が、状況に応じてＩＣカードＯＳプログラム３３４１によって既定された機能を変化させたり、新しい機能を追加するために使用する書き換え可能な情報である。各ＩＣカードアプリケーション（１７３、１７４、１７５など）は、さらに３つの構成要素からなる。図３３では、アプリケーションＡ１７３についてそれらの構成要素（アプリケーションＡメインプログラム３３２０、モジュールロード領域１８５、モジュール管理情報１８４）を示す。アプリケーションＡメインプログラム３３２０は、アプリケーションＡ１７３の４つの機能ブロック（図２７における、コマンド解釈部１８１、モジュール実行部１８２、モジュールロード部１８３、認証部１８６）を含み、ＩＣカードＯＳ１７２上で実行されるプログラムである。モジュールロード領域１８５は、アプリケーションＡ用コマンド処理モジュール群３１１１の７モジュール（３２０１～３２０７）から実行したいものをロードするための領域であり、１つ以上のモジュールを格納できるサイズが確保されている。例えば、３つの領域（第１領域３３３１、第２領域３３３２、第３領域３３３３）が用意されている。この場合、７モジュールのうち最大３モジュールまでロードすることができる。よって、ＥＥＰＲＯＭ１６２において、アプリケーションＡ１７３に消費される領域サイズをできるだけ小さくしたいならば、モジュールロード領域１８５は、１モジュールを格納できるサイズを確保すればよい。次に、モジュール管理情報１８４は、各コマンド処理モジュール（３２０１～３２０７）をモジュールロード領域１８５にロードするときなどに使用されるデータであり、アプリケーションＡ１７３内でその内容を参照したり更新したりすることができる。また、ＭＭＣ１１０内のコントローラチップ１

２０がその内容を読み出すこともできる。さらに、アプリケーションＡ１７３の改訂などにおいて、そのコマンド処理モジュール（３２０１～３２０７）を更新する際には、コントローラチップ１２０がその内容を書き換えることもできる。

【００８９】図３５は、アプリケーションＡ１７３のモジュール管理情報１８４の内容を示している。モジュール管理情報１８４は、ロード管理情報３５１０とロード領域情報３５２０から構成される。ロード管理情報３５１０は、アプリケーションＡ用コマンド処理モジュール群３１１１の各モジュール（３２０１～３２０７）に関する情報であり、ロード領域情報３５２０は、モジュールロード領域１８５の各領域（３３３１～３３３３）に関する情報である。ロード管理情報３５１０は、各モジュールの識別番号（番号３５１１）とコマンド処理の内容（処理内容３５１２）との対応関係を示す情報を含む。コントローラチップ１２０がモジュールをロードする際には、番号３５１１をコマンド処理モジュール１８７に付してＩＣカードチップ１５０にロードする。なお、処理内容３５１２に記載するデータとして、ホスト機器２２０から受信したセキュリティ処理要求コマンドのヘッダ部分（コマンドクラス番号や命令コード番号など）を利用すれば、コントローラチップ１２０によるモジュール選択がしやすい。ロード管理情報３５１０は、各モジュールのプログラムサイズ（サイズ３５１３）（単位はバイト）を含み、モジュールロード領域１８５の各領域に格納可能なサイズかどうかを判定できるようにしている。ロード管理情報３５１０は、各モジュールのロード状態（状態３５１４）を含む。図３５では、例えば、第１ＰＩＮ検証処理モジュール３２０１がモジュールロード領域１８５の第２領域３３３２にロードされており、乱数発生処理モジュール３２０３はモジュールロード領域１８５にロードされていないことを示している。ロード管理情報３５１０は、その時点でロードされている各モジュールの改訂番号（使用版３５１５）を含む。ロード管理情報３５１０は、各モジュールの最新の改訂番号（最新版３５１６）を含む。モジュールの更新時にＭＭＣ１１０のホスト機器２２０から最新の改訂番号が知らされ、そのモジュールの最新版３５１６の値がその番号に更新される。その際、更新するモジュール以外のモジュールの最新版３５１６の値も同時に更新してもよい。ロード管理情報３５１０は、ロード条件３５１７を含む。ロード条件３５１７は、モジュールをロードするときに、ロードを許可するかどうかをアプリケーションＡ１７３が判定するための条件であり、その使用版３５１５や最新版３５１６の値に基づく。図３５では、例えば、署名検証処理モジュール３２０２は改訂番号に関係なくロードを許可し、暗号化処理モジュール３２０４は最新版のみロードを許可し、復号・比較処理モジュール３２０５は使用版３５１５の値が１．０以上ならばロードを許可することを意味する。次に、ロード領域情報３５２０は、モジュールロード領域１８５の各領域（３３３１～３３３３）の容量サイズ３５２１（単位はバイト）を含む。これは、各領域（３３３１～３３３３）が、サイズ３５２１の値より大きなサイズのモジュールをロードできないことを示すための情報である。例えば、第２領域３３３２には３２０バイトの容量しかないため、復号・比較処理モジュール３２０５（サイズが３８８バイト）はロードすることが許可されない。ロード領域情報３５２０は、領域識別番号３５２２を含む。これは、コントローラチップ１２０が、コマンド処理モジュール１８７をロードする領域を指定するのに使用する。

【００９０】図３４は、コントローラチップ１２０によるコマンド処理モジュール１８７のロード、およびＩＣカードチップ１５０によるセキュリティコマンドの実行の手順を示すフローチャートである。まず、コントローラチップ１２０はモジュール管理情報１８４を使用してモジュールをロードするかを判断する（３４０１）。このステップ３４０１は、図１３ではステップ１３０６に相当する。コントローラチップ１２０が、必要なモジュールがすでにロードされていることを知っているならば、ステップ３４１２に移ってよい。これは、図１３ではステップ１３０６から１３０８への遷移に相当する。一方、コントローラチップ１２０が、モジュール管理情報１８４を使用してモジュールをロードする必要があるかを判断したいならば、ステップ３４０２に遷移する。これは、図１３ではステップ１３０７（環境設定の実行）への遷移に相当する。ステップ３４０２では、コントローラチップ１２０は、環境設定のためのＩＣカードコマンドとして、モジュール管理情報１８４のリードコマンドをＩＣカードチップ１５０に発行する。ＩＣカードチップ１５０のＩＣカードＯＳ１７２は、アプリケーションＡ１７３にこのコマンドの処理権を渡す。アプリケーションＡ１７３は、コマンド解釈部１８１でこのコマンドを解釈し、ＩＣカードＯＳ１７２を通してモジュール管理情報１８４をコントローラチップ１２０へ送信する（３４０３）。コントローラチップ１２０は、モジュール管理情報１８４の内容を調べ、所望のコマンド処理モジュール１８７が、モジュールロード領域１８５にロードされているかを確認する（３４０４）。そして、存在するならばステップ３４１２に移り、存在しなければステップ３４０６に移る（３４０５）。コントローラチップ１２０は、フラッシュメモリチップ１３０のモジュールデータ領域３１１０から所望のコマンド処理モジュール１８７（ここでは３２０１から３４０７のいずれか）を読み出す（３４０６）。そして、コントローラチップ１２０は、環境設定のためのＩＣカードコマンドとして、モジュールロードコマンドをＩＣカードチップ１５０に発行し、読み出したモジュール、モジュール識別

番号（ここでは１～７の範囲）、ロード領域番号（ここでは１～３の範囲）を送信する（３４０７）。ＩＣカードチップ１５０のＩＣカードＯＳ１７２は、アプリケーションＡ１７３にこのコマンドの処理権を渡す。アプリケーションＡ１７３は、コマンド解釈部１８１でこのコマンドを解釈し、認証部１８６において、受信したコマンド処理モジュール１８７が適正かを判定する（３４０８）。具体的には、コマンド処理モジュール１８７にかけられた暗号を解き、実行可能なデータ形式であるか、サイズが適当であるかなどを調べる。そのため、認証部１８６は、モジュールにかけられた暗号を解くためのモジュール鍵を持つ。望ましくは、モジュールの改ざん防止のために電子署名を適用する。すなわち、アプリケーションＡ１７３に固有の秘密鍵（アプリケーション秘密鍵）とそれに対応する公開鍵（アプリケーション公開鍵）を用意し、コマンド処理モジュール１８７、または、その暗号化前の実行可能形式データにアプリケーション秘密鍵によって電子署名を付けておき、認証部１８６でアプリケーション公開鍵によってその署名を検証するようにする。この場合、認証部１８６は、アプリケーション公開鍵も持つ。ステップ３４０８において、コマンド処理モジュール１８７が適正でないならば、ステップ３４１１に移り、ＩＣカードチップ１５０は、ロード結果が失敗であることをコントローラチップ１２０に返す。コマンド処理モジュール１８７が適正ならば、アプリケーションＡ１７３のモジュールロード部１８３は、モジュールロード領域１８５の指定された領域（３３３１、３３３２、３３３３のいずれか）にライトし（３４０９）、モジュール管理情報１８４の状態３５１４を更新する（３４１０）。そして、ＩＣカードチップ１５０は、ロード結果が成功であることをコントローラチップ１２０に返す（３４１１）。そして、コントローラチップ１２０は、ステップ３４１２～３４１６にわたって、アプリケーションＡ１７３によるセキュリティコマンド処理を実行する。これは、図１３のステップ１３０９に相当する。まず、コントローラチップ１２０は、ＩＣカードチップ１５０にセキュリティコマンドを発行する（３４１２）。セキュリティコマンドとは、図２９におけるステップ２９０１、２９０８、２９１０、２９１３、図３０におけるステップ３００３、３００９、３０１２のいずれかで発行するコマンドに応じて、ＭＭＣ１１０内部で発行されるＩＣカードコマンドである。ＩＣカードチップ１５０内のアプリケーションＡ１７３のコマンド解釈部１８１は、モジュール管理情報１８４により、そのコマンドに対応するコマンド処理モジュール１８７がモジュールロード領域１８５に存在するかを調査する（３４１３）。存在しないならば、処理結果が失敗であることをコントローラチップ１２０に返し（３４１５）、ステップ３４１６に移る。存在するならば、モジュール実行部１８２は、モジュールロード領域１８５からそのコマンド処理モジュール１８７を読み、セキュリティコマンドを処理する（３４１４）。そして、その処理結果をコントローラチップ１２０に返す（３４１５）。コントローラチップ１２０は、その処理結果を受信する（３４１６）。以上が、コントローラチップ１２０によるコマンド処理モジュール１８７のロード、およびＩＣカードチップ１５０によるセキュリティコマンドの実行手順である。

【００９１】図３６は、フラッシュメモリチップ１３０に格納されたアプリケーションＡ１７３のためのコマンド処理モジュール１８７を、ホスト機器２２０によって更新する手順を示すフローチャートである。ここで、ホスト機器２２０は、アプリケーションＡ１７３のモジュール更新することを許可されており、上述のアプリケーション秘密鍵を持っているものとする。また、この手順においてホスト機器２２０からＭＭＣ１１０に送信される新しいコマンド処理モジュールは、その識別番号、サイズ、改訂番号（それぞれ、図３５における番号３５１１、サイズ３５１３、最新版３５１６に登録されるべき情報）が含まれている。また、この手順に先だって、ＩＣカードチップ１５０では、アプリケーションＡ１７３が選択されているものとする。まず、ホスト機器２２０は暗号化乱数発生コマンドを発行する（３６０１）。ＭＭＣ１１０は、ＩＣカードチップ１５０のアプリケーションＡ１７３の認証部１８６において、乱数を発生させ（３６０２）、その乱数を上述のアプリケーション公開鍵で暗号化させ、暗号化乱数をホスト機器２２０に送信する（３６０３）。ホスト機器２２０は、アプリケーション秘密鍵で暗号化乱数を復号し、乱数を復元する（３６０４）。そして、ホスト機器２２０は、その乱数と新しいコマンド処理モジュールを連結したデータを作り、そのデータにアプリケーション秘密鍵で電子的に署名する（３６０５）。ホスト機器２２０は、ＭＭＣ１１０に検証・ライトコマンドを発行して、その署名付き連結データを送信する（３６０６）。ＭＭＣ１１０は、ＩＣカードチップ１５０のアプリケーションＡ１７３の認証部１８６において、アプリケーション公開鍵で連結データの署名を検証させる（３６０７）。ステップ３６０８では、連結データ内の乱数がステップ３６０２で発生した乱数と一致するかを比較する。ステップ３６０７での検証が成功し、かつそれらの乱数が一致したならば、受信した新しいコマンド処理モジュールは正しいものであることが証明されたので、新しいコマンド処理モジュールを、アプリケーションＡ用コマンド処理モジュール群３１１１の中の同じ識別番号のコマンド処理モジュールに上書きする（３６０９）。さらに、ＩＣカードチップ１５０にＩＣカードコマンドを送ることによって、アプリケーションＡ１７３内のモジュール管理情報１８４の内容（図３５における番号３５１１、サイズ３５１３、最新版３５１６に登録される情報）も更新する。そして、

45

ホスト機器２２０は更新処理を終了する（３６１０）。一方、ステップ３６０７での検証が失敗するか、または乱数が一致しないならば、コマンド処理モジュールやモジュール管理情報１８４の内容の更新は実行せず、ホスト機器２２０は更新処理を終了する（３６１０）。

【００９２】図３７は、図３４や図３６の手順で登場したアプリケーションに固有な３種類の鍵（モジュール鍵、アプリケーション秘密鍵、アプリケーション公開鍵）を説明したものである。モジュール鍵３７０１は、コマンド処理モジュールにかけられた暗号を解き、実行可能形式に復元するための対称鍵である。アプリケーション秘密鍵３７０２は、アプリケーション発行者が厳重に管理すべき秘密鍵であり、コマンド処理モジュールの更新手順において乱数の復号、署名の作成に使用される。アプリケーション公開鍵３７０３は、アプリケーション秘密鍵３７０２に対応する公開鍵であり、コマンド処理モジュールのロード手順・更新手順において乱数の暗号化、署名の検証に使用される。

【００９３】上記では、コマンド処理モジュール１８７が１つのセキュリティコマンドを処理するプログラムモジュールであると説明されているが、複数のセキュリティコマンドを１つのコマンド処理モジュール１８７が処理してもよい。

【００９４】本発明によれば、ＩＣカードチップによるセキュリティ処理に必要なデータやＩＣカードチップを管理するための情報をフラッシュメモリに保持し、ＩＣカードチップ内のセキュリティ処理プログラムが消費するメモリ資源を節約し、より多くのプログラムを登録することができる。したがって、ユーザの利便性を向上させることができる。

【００９５】尚、１つのMMC110が、複数のＩＣカードチップ150を備えてもよい。複数のＩＣカードチップ150の個々は、異なるアプリケーションプログラムを実行してもよい。例えば、３つのＩＣカードチップ150のうちの第１のＩＣカードチップ150が、アプリケーションプログラムA173を実行し、３つのＩＣカードチップ150のうちの第２のＩＣカードチップ150が、アプリケーションプログラムB174を実行し、３つのＩＣカードチップ150のうちの第３のＩＣカードチップ150が、アプリケーションプログラムC175を実行するのが好ましい。さらに、複数のＩＣカードチップ150の個々は、異なる者によって認可又は発行されてもよい。例えば、第１のＩＣカードチップ150は、アプリケーションプログラムA173を用いて銀行取引サービスを実現する例えば銀行によって認可又は発行され、第２のＩＣカードチップ150は、アプリケーションプログラムB174を用いてクレジット決済サービスを実現する例えばクレジット会社によって認可又は発行されるのが好ましい。第３のＩＣカードチップ150は、アプリケーションプログラムC175を用いてコンテンツ配信サービスを実現する例えばコンテンツプロバイダーによって認可又は発行されてもよいし、第３者（例えば、ＩＣカードチップ150のセキュリティを認可又は保証する機関）によって認可又は発行されてもよい。この場合、銀行、クレジット会社、コンテンツプロバイダーや第３者によって認可又は発行された後、MMC110に搭載するのが好ましい。つまり、認可又は発行されたＩＣカードチップ150を、MMC110上に固定し、ＩＣカードチップ150内のインターフェース150～157と、ＩＣカードＩ／Ｆ制御回路128、ＣＬＫ2制御回路127、ＶＯＣ2制御回路126、ＧＮＤ1とを電気的に接続する。

46

【００９６】これまでに述べたＭＭＣ１１０のセキュリティ処理においては、ＩＣカードチップ１５０に送信するＩＣカードコマンドによって扱うべきデータサイズが、ＩＣカードチップ１５０内部で利用可能なワークメモリの空き容量を越えてしまうと、セキュリティ処理が実行不能になってしまう可能性がある。例えば、ワークメモリ（例えば、RAM160のＩＣカードＯＳワーク領域3351やアプリケーションワーク領域3352等）の空き容量が１００バイトのときに、ＩＣカードコマンドにより２００バイトの暗号文の復号が要求されても処理できない。このような問題を解決するために、ＭＭＣ１１０は、ＩＣカードチップ１５０が状況（例えば、ＩＣカードコマンドによって要求される処理に必要なメモリ量やＩＣカードコマンドによって処理すべきデータ量）に応じて外部に対して能動的に要求を出す機能を搭載する。以下、その機能について説明する。

【００９７】上記の問題を解決するには、まず、ＩＣカードチップ１５０は、セキュリティのためのコマンドを受け取り、その後処理すべきデータ量を算出し、その処理すべきデータ量がその時点でのワークメモリの空き容量より大きい又は以上であるかを判断するとよい。比較すべき空き容量は、コマンドを受け取ってからその都度計算してもよいし、常時計算しておいてレジスタに保持しておき、その値を利用してもよい。もしそれが空き容量より大きい又は以上でなければ（空き容量より小さい又は以下であれば）、コマンドで指示された処理を実行し、コントローラチップ１２０にその処理結果を応答として返す。一方、もしそれが空き容量より大きい又は以上であるならば、状況に応じた要求を含む応答をコントローラチップ１２０に返す。その要求の内容としては、例えば、「たったいま送信したデータをフラッシュメモリチップ１３０にライトし、１２８バイト単位に分割して送れ」とか、「ホスト機器２２０は、処理すべきデータを１２８バイト単位に分割して再送信せよ」といったものが考えられる。コントローラチップ１２０が、このような要求にしたがって適当な処理を実行すれば、ＩＣカードチップ１５０によるセキュリティ処理が実行不能になることを防止することができる。

【００９８】このような機能は、図２３を用いて説明し

たコンテンツ配信の例にも適用することができる。暗号化されたコンテンツ２３１４をセッション鍵によって復号する手順２３２４において、コンテンツ２３１４のデータサイズがワークメモリの空き容量より大きい又は以上であったならば、ＩＣカードチップ１５０は「ホスト機器２２０は、コンテンツ２３１４を１２８バイト単位に分割して再送信せよ」などの要求を含んだ応答を返す。コンテンツ２３１４のデータを一括して受け取って、その復号処理を実行し、処理結果（成功か失敗か等）を含んだ応答を返してもよい。これにより、ＩＣカードチップ１５０による復号処理が実行不能になることを防止することができる。ＩＣカードチップ１５０は、これに加えて、さらに「コントローラチップ１２０は、復号したコンテンツをフラッシュメモリチップ１３０にライトせよ」という要求を含んだ応答を返してもよい。そうすれば、ホスト機器２２０が、コンテンツをフラッシュメモリチップ１３０にライトするため、ＭＭＣ１１０に対してライトコマンドを送信する手間を省くこともできる。

【００９９】以下、ＩＣカードチップ１５０がコントローラチップ１２０を通じて要求する何らかの処理のことを、「外部処理」と一般化して呼ぶこととする。

【０１００】図３８を用いてＭＭＣ１１０が外部処理を実行する手順を詳細に説明する。先ず、コントローラチップ１２０はセキュリティ処理のためのＩＣカードコマンドを発行する（３８０１）。ＩＣカードチップ１５０はそのコマンドを解析し、外部処理が必要かを判断する（３８０２）。例えば、前述例のように、ＩＣカードコマンドで入力されたデータの量とＩＣカードチップ１５０内のワークメモリの空き容量とを比較し、データ量がメモリ容量より大きい又は以上であれば外部処理が必要と判断し、データ量がメモリ容量より小さい場合は外部処理が不要と判断する。手順３８０２の結果、外部処理が必要ならば、外部処理要求コードを含むレスポンスを送信する（３８０３）。外部処理要求コードとは、コントローラチップ１２０に外部処理を要求したいということを告知するための符号である。ＩＣカードチップ１５０が準拠するＩＳＯ７８１６のコマンド規格によれば、Ｉ／Ｏ端子１５７からシリアルデータとして出力される、ＩＣカードコマンドのレスポンス信号は、コマンド処理状況を示す「ステータスワード」と呼ばれる例えば２バイトの符号化データをその末尾部（又は前頭部）に含む。これは主に、コマンド処理結果がエラー（６Ｘｈ、ＸＸｈ（Ｘはエラー内容に依存））であるか、成功（９０ｈ、００ｈ）であるかをＩＣカードチップ外部に示すために使用される。外部処理要求コードは、このステータスワードを利用するのが好ましい。すなわち、ステータスワード＝９１ｈ、ＹＹｈを外部処理要求コードと定義する。ここで、ＹＹｈは、外部処理の内容を示したデータの長さがでＹＹｈ（１６進数）バイトであることを予告している。手順３８０３の次に、コントローラチップ１２０は、レスポンスを受信し、その受信したレスポンスを分析し、外部処理要求コードが含まれるかを調べる（３８０４）。外部処理要求コードが含まれているならば、外部処理内容を読み出すＩＣカードコマンドを発行する（３８０５）。そして、ＹＹｈバイトの外部処理内容を含む応答データを待つ。ＩＣカードチップ１５０がＹＹｈバイトの外部処理内容を含む応答データを送信する（３８０６）と、コントローラチップ１２０は受信した外部処理内容を分析する（３８０７）。そして、コントローラチップ１２０は、外部処理内容に応じて、フラッシュメモリチップ１３０またはホスト機器２２０に対して、上の例で記したような外部処理を指示する（３８０８）。フラッシュメモリチップ１３０またはホスト機器２２０は、指示された外部処理を実行する（３８０９）。その後、コントローラチップ１２０は、その外部処理の結果を示すデータを作成し（３８１０）、ＩＣカードチップ１５０に結果通知コマンドを発行して外部処理の結果を送信する（３８１１）。ＩＣカードチップ１５０は、外部処理の結果を分析し、正しく処理されたか、または、さらなる外部処理が必要かどうかを判断する（３８１２）。そして、その判断に応じたレスポンスを送信する（３８１３）。コントローラチップ１２０は、そのレスポンスを分析する（３８１４）。そのレスポンス中のステータスワードがエラー（６Ｘｈ、ＸＸｈ）であれば、外部処理はエラーであることを意味する。それが成功（９０ｈ、００ｈ）であれば、外部処理は正常に終了したことを意味する。もし、ステータスワードが外部処理要求コード（９１ｈ、ＹＹｈ）であれば、コントローラチップ１２０は、手順３８０５に戻って、次に要求される外部処理内容を読み出す手順から再び始める。

【０１０１】なお、手順３８０６において送信する外部処理内容の表記法としては、コントローラチップ１２０とＩＣカードチップ１５０との間であらかじめ定義された符号化ルールを利用することが好ましい。符号化ルールとしては、例えば、「フラッシュメモリチップ１３０」を０１ｈ、「ＩＣカードチップ１５０」を０２ｈ、「直前に送信されたデータ」を１１ｈ、「ライト処理せよ」を２２ｈ、「１２８（８０ｈ）バイト単位に分割して送信せよ」を２３８０ｈなどとあらかじめ定義しておく。この場合、一例として、外部処理内容データに０１ｈ、１１ｈ、２２ｈ、０２ｈ、１１ｈ、２３ｈ、８０ｈという７バイトの符号を設定することは、「たったいま送信したデータをフラッシュメモリチップ１３０にライトし、それを１２８バイト単位に分割してＩＣカードチップ１５０に送れ」という内容を意味している。このような符号化ルールは、コントローラチップ１２０内部に格納しておいたものを参照してもよいし、フラッシュメモリチップ１３０に格納しておき、必要に応じてそこか

ら読み出して参照してもよい。

【０１０２】また、手順３８０６において送信する外部処理内容が、フラッシュメモリチップ１３０に対するリード・ライト処理である場合は、外部処理内容のデータ形式をＭｕｌｔｉＭｅｄｉａＣａｒｄ仕様に定義されたリード・ライトコマンドと同じ形式にしてもよい。これは、ホスト機器２２０がＭＭＣ１１０に対して発行するコマンドと同じ構造のデータ形式となるため、コントローラチップ１２０内部にあるＭＭＣコマンドの分析手段に外部処理内容のデータを直接入力するだけで外部処理（フラッシュメモリに対するリード・ライト処理）が実行できるため、効率的な処理が実現できる。

【０１０３】また、手順３８０６において送信する外部処理内容が、ホスト機器２２０に対して要求する処理である場合は、外部処理内容のデータ形式をＭｕｌｔｉＭｅｄｉａＣａｒｄ仕様に定義されたコマンドのレスポンスと同じ形式にしてもよい。そうすれば、ホスト機器２２０内部にあるＭＭＣレスポンスの分析手段に外部処理内容のデータを直接入力するだけで外部処理が実行できるため、効率的である。

【０１０４】ホスト機器２２０が通信機能を持つような機器であるならば、ホスト機器２２０に対する外部処理として、「遠隔地のサーバにデータを送信せよ」といった内容にすることも可能である。これにより、ホスト機器２２０内部にあるサーバ通信プログラムの一部を削減できるため、ホスト機器２２０内部のプログラムメモリを節約できる。

【０１０５】ホスト機器２２０がＸＭＬ（ｅＸｔｅｎｓｉｂｌｅ　Ｍａｒｋｕｐ　Ｌａｎｇｕａｇｅ：拡張可能なマーク付け言語）に対応したブラウザ機能を持つような機器であるならば、ホスト機器２２０に対する外部処理として、「データをブラウザによってディスプレイに表示せよ」とった内容も可能である。このとき、ＩＣカードチップ１５０が送信する表示データをＸＭＬ言語で書かれたものにすれば、ホスト機器２２０内部で表示データ変換をおこなわなくてもよいため、効率的な表示処理が実現できる。

【０１０６】本発明によれば、ＩＣカードチップが他のチップに対して能動的に処理を要求することにより、ＩＣカードチップによるセキュリティ処理において、フラッシュメモリやホスト機器など外部のデバイスを有効に活用して、より大きなサイズのデータを処理することができるため、利便性の高いセキュリティシステムを実現することができる。

【０１０７】

【発明の効果】本発明によれば、ＩＣが実行するためのプログラムやデータ、ＩＣを管理するための情報をＩＣ外部の不揮発性メモリに保持するため、ＩＣ内の記憶容量（例えば、ＲＯＭやＥＥＲＯＭ）が小さい場合にも、ＩＣが多くの処理を実行できるという効果を奏する。

【０１０８】本発明によれば、ＩＣが実行する処理の一部をＩＣ外部のコントローラが実行するため、ＩＣ内の記憶容量（例えば、ＲＡＭ）が小さい場合にも、ＩＣが多くの処理を実行できるという効果を奏する。

【図面の簡単な説明】

【図１】　本発明を適用したＭＭＣの内部構成を示す図である。

【図２】　本発明を適用したＭＭＣのホスト機器の内部構成、およびホスト機器とＭＭＣとの接続状態を示す図である。

【図３】　ＩＣカードチップのコールドリセット時の信号波形を示す図である。

【図４】　ＩＣカードチップのウォームリセット時の信号波形を示す図である。

【図５】　ＩＣカードチップのＩＣカードコマンド処理時の信号波形を示す図である。

【図６】　ＩＣカードチップの非活性化時の信号波形を示す図である。

【図７】　ホスト機器によるＭＭＣへのアクセスを示したフローチャートである。

【図８】　ＩＣカード制御パラメータとそれに対応するＩＣカードへの処理内容を示す表である。

【図９】　ＩＣカードチップに対する第１次ＩＣカード初期化の詳細なフローチャートである。

【図１０】　ＩＣカードチップに対する第２次ＩＣカード初期化の詳細なフローチャートである。

【図１１】　非活性状態のＩＣカードチップに対するＩＣカード初期化時の信号波形を示す図である。

【図１２】　活性状態のＩＣカードチップに対するＩＣカード初期化時の信号波形を示す図である。

【図１３】　ＩＣカードチップによるセキュリティ処理の詳細なフローチャートである。

【図１４】　セキュリティ処理要求ライトコマンドを処理するときの信号波形とフラッシュメモリチップアクセスを示す図である。

【図１５】　ＩＣカードチップによるセキュリティ処理実行時の信号波形とフラッシュメモリチップアクセスの一例を示す図である。

【図１６】　セキュリティ処理結果リードコマンドを処理するときの信号波形とフラッシュメモリチップアクセスを示す図である。

【図１７】　インタフェース直通モードにおけるＭＭＣ外部端子とＩＣカードチップ外部端子の対応関係を示す図である。

【図１８】　インタフェース直通モードへ移行する処理とインタフェース直通モードから復帰する処理のフローチャートである。

【図１９】　インタフェース直通モードへ移行する処理時の信号波形を示す図である。

51

【図２０】　インタフェース直通モードから復帰する処理時の信号波形を示す図である。

【図２１】　フラッシュメモリチップの内部構成を示す図である。

【図２２】　本発明を適用したＭＭＣの内部構成を簡単に示す図である。

【図２３】　本発明を適用したＭＭＣをコンテンツ配信に応用した例を示す図である。

【図２４】　本発明を適用したＳＤカードの内部構成を簡単に示す図である。

【図２５】　本発明を適用したメモリースティックの内部構成を簡単に示す図である。

【図２６】　本発明のＩＣカードチップの内部構成を示す図である。

【図２７】　本発明を適用したＭＭＣの内部構成を示し、特にＩＣカードチップの機能的構成の詳細を含む図である。

【図２８】　本発明を適用したＭＭＣを銀行取引サービスに応用した場合の、システム構成を示す図である。

【図２９】　本発明を適用したＭＭＣを銀行取引サービスに応用した場合の、取引実行時のフローチャートの前半である。

【図３０】　本発明を適用したＭＭＣを銀行取引サービスに応用した場合の、取引実行時のフローチャートの後半である。

【図３１】　フラッシュメモリチップの内部構成を示し、特にＩＣカード環境設定情報領域の詳細を含む図である。

52

【図３２】　コマンド処理モジュール群の一例を示す図である。

【図３３】　ＩＣカードチップの中のＲＯＭ、ＲＡＭ、ＥＥＰＲＯＭの内部構成を示す図である。

【図３４】　コマンド処理モジュールのロード手順およびセキュリティコマンドの実行手順のフローチャートである。

【図３５】　モジュール管理情報の詳細および具体例を示す図である。

【図３６】　コマンド処理モジュールの更新手順のフローチャートである。

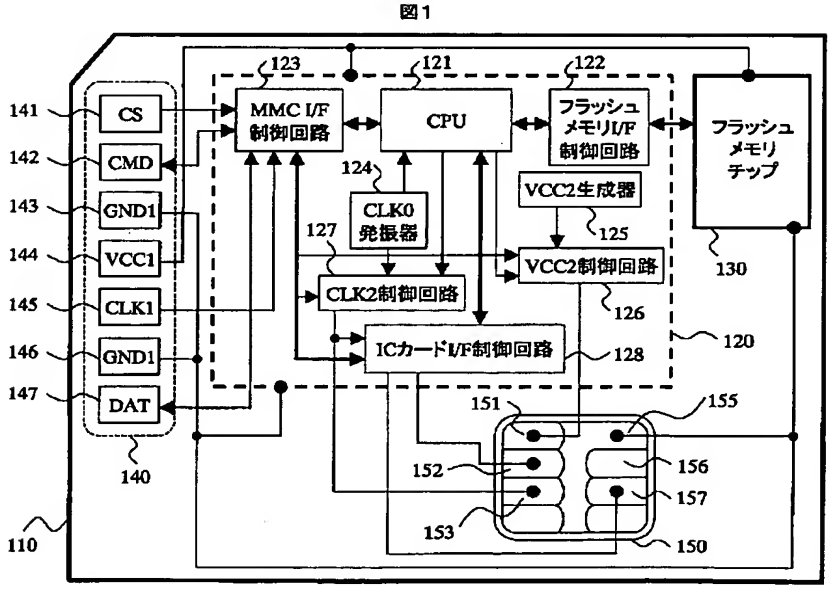
【図３７】　コマンド処理モジュールのロードや更新に使用する鍵の説明を示す図である。

【図３８】　ＩＣカードチップの要求に従い、コントローラチップがフラッシュメモリチップやホスト機器に処理を実行させる手順を示すフローチャートである。

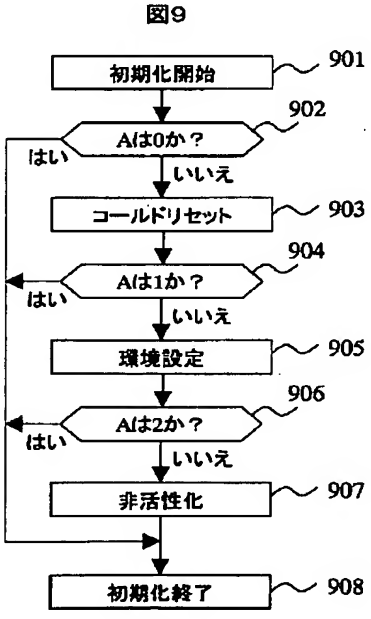
【符号の説明】

１１０…ＭＭＣ、１２０…コントローラチップ、１４０…ＭＭＣ外部端子、１５０…ＩＣカードチップ、１５１…ＶＣＣ２端子、１５２…ＲＳＴ端子、１５３…ＣＬＫ２端子、１５５…ＧＮＤ２端子、１５６…ＶＰＰ端子、１５７…Ｉ／Ｏ端子、２２０…ホスト機器、１４０５…ライトコマンド発行、１９０６…モード移行時刻、２００３…モード復帰時刻。

【図１】

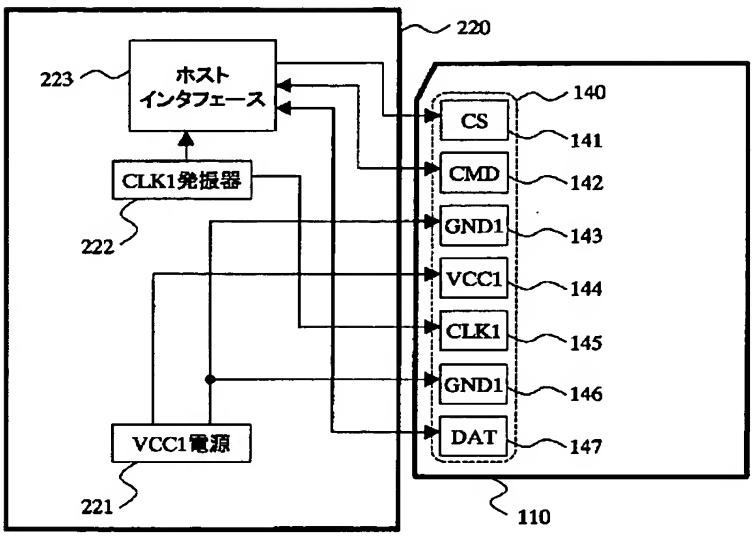


【図９】



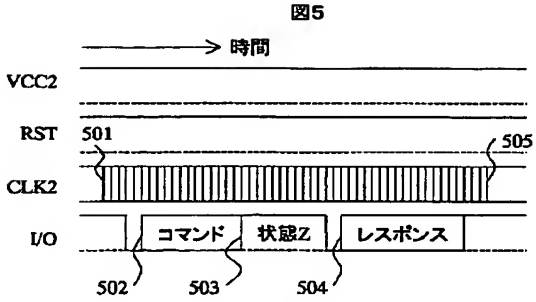
【図２】

図2



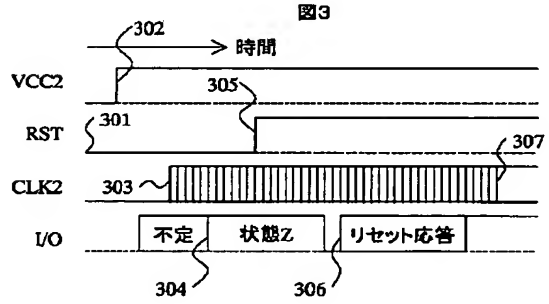
【図５】

図5



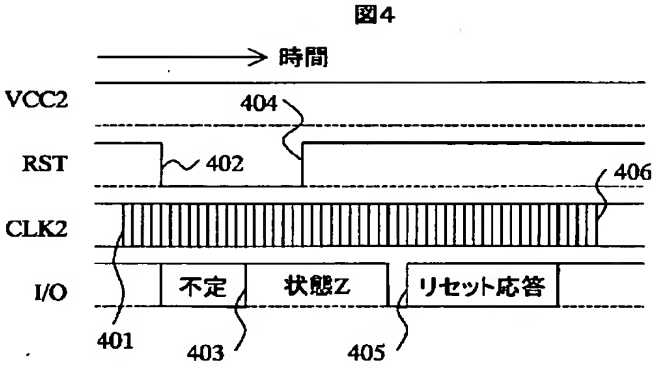
【図３】

図3



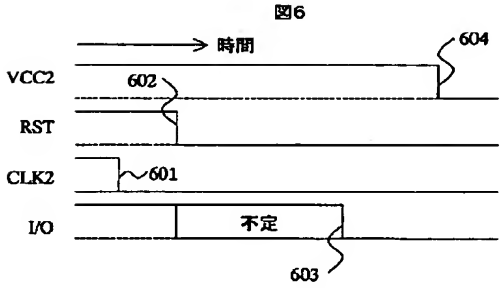
【図４】

図4



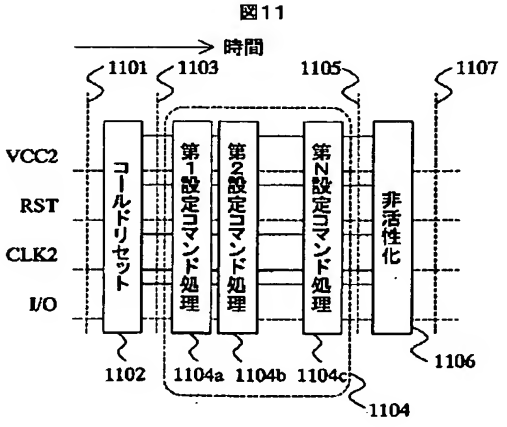
【図６】

図6



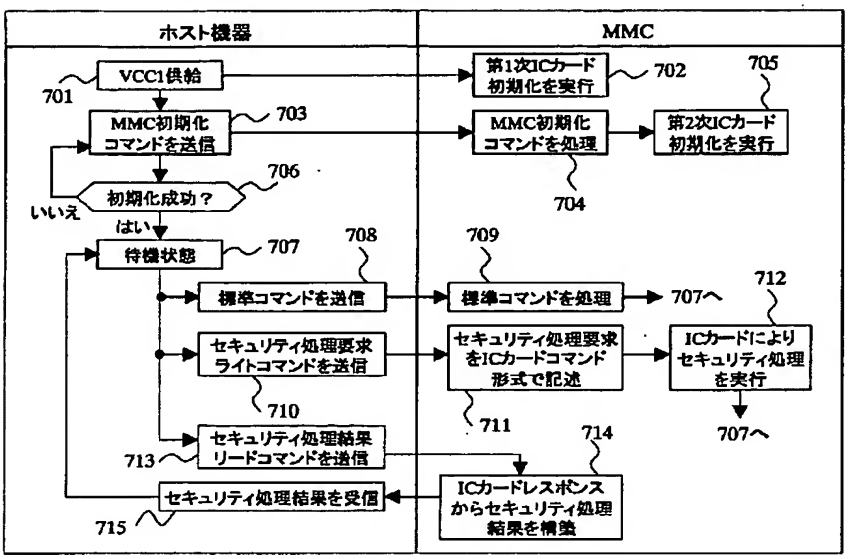
【図１１】

図11



【図７】

図7



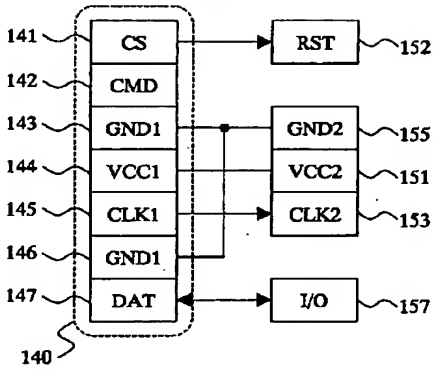
【図８】

図8

ICカード制御パラメータ		ICカードに対する処理
A=0		MMCのパワーオン時に，何もしない
A=1		MMCのパワーオン時に，リセット
A=2		MMCのパワーオン時に，リセットと環境設定
A=3		MMCのパワーオン時に，リセットと環境設定し，非活性化
B=0		MMCの初期化時に，何もしない
B=1	C=1	MMCの初期化時に，リセット
	C=2	MMCの初期化時に，リセットと環境設定
	C=3	MMCの初期化時に，リセットと環境設定し，非活性化
B=2	C=2	MMCの初期化時に，環境設定
	C=3	MMCの初期化時に，環境設定し，非活性化
B=3		MMCの初期化時に，活性状態ならば，非活性化
D=0		セキュリティ処理後に，非活性化しない
D=1		セキュリティ処理後に，非活性化する

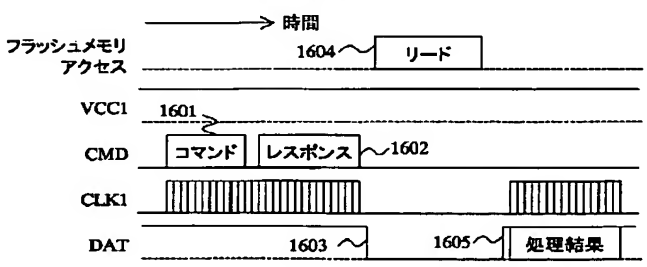
【図１７】

図17



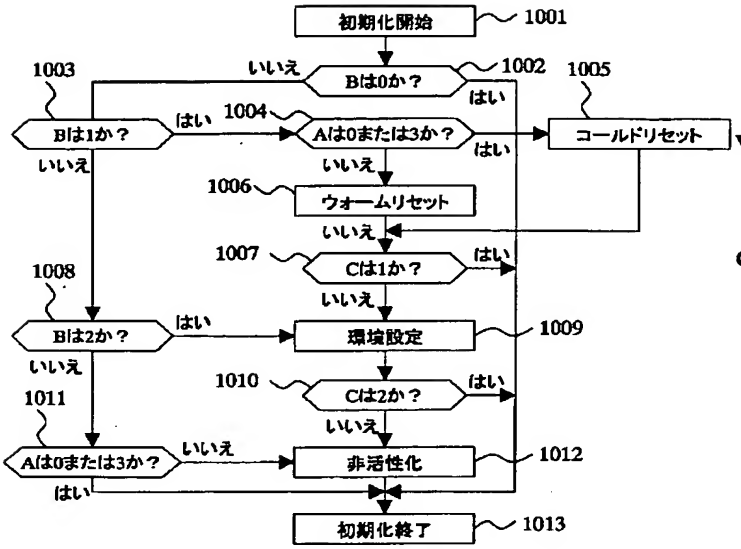
【図１６】

図16



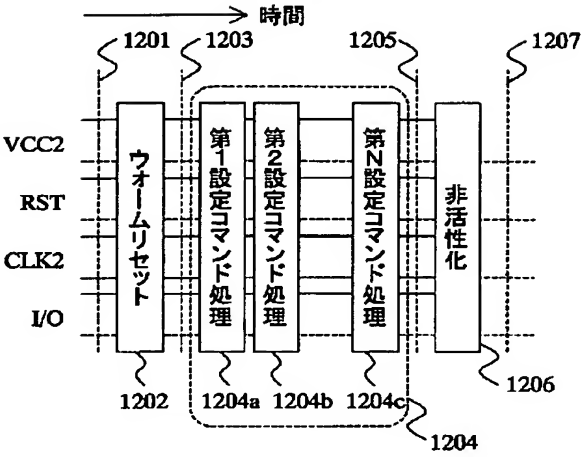
【図１０】

図10



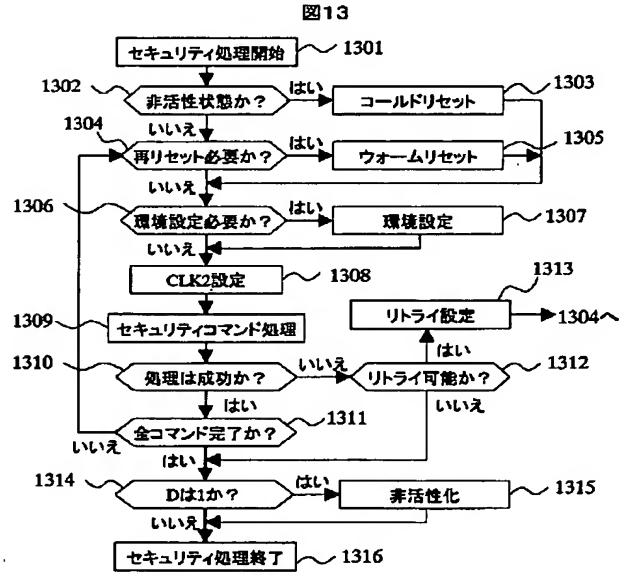
【図１２】

図12



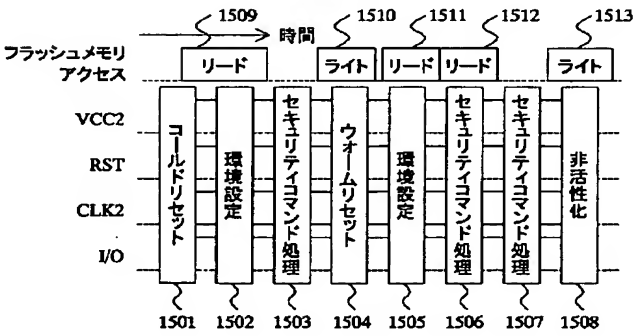
【図１３】

図13



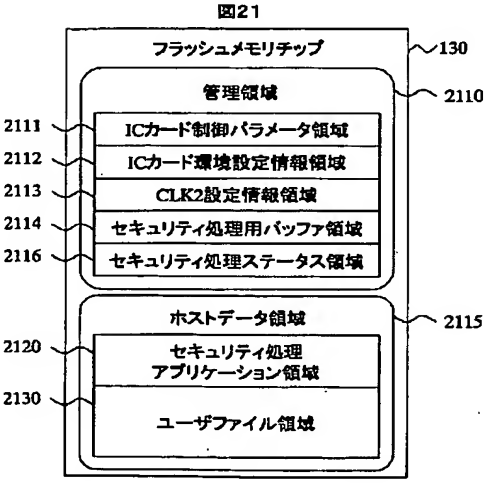
【図１５】

図15



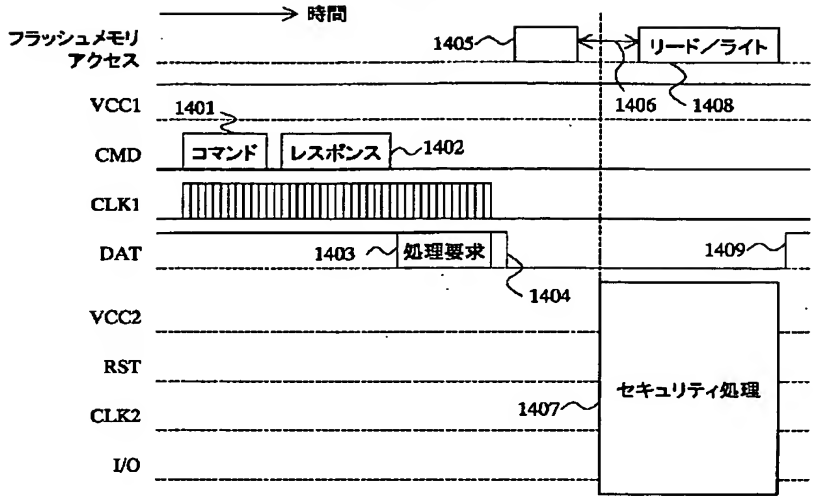
【図２１】

図21



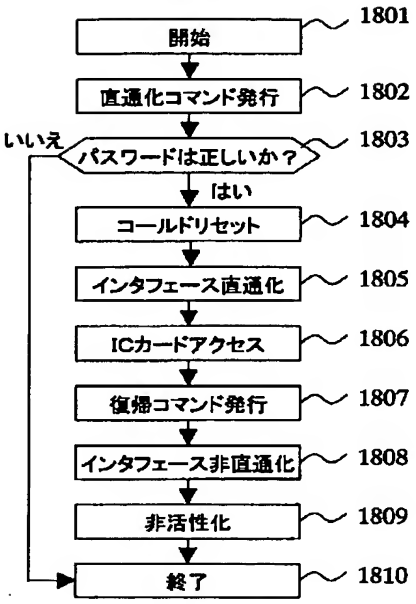
【図１４】

図14



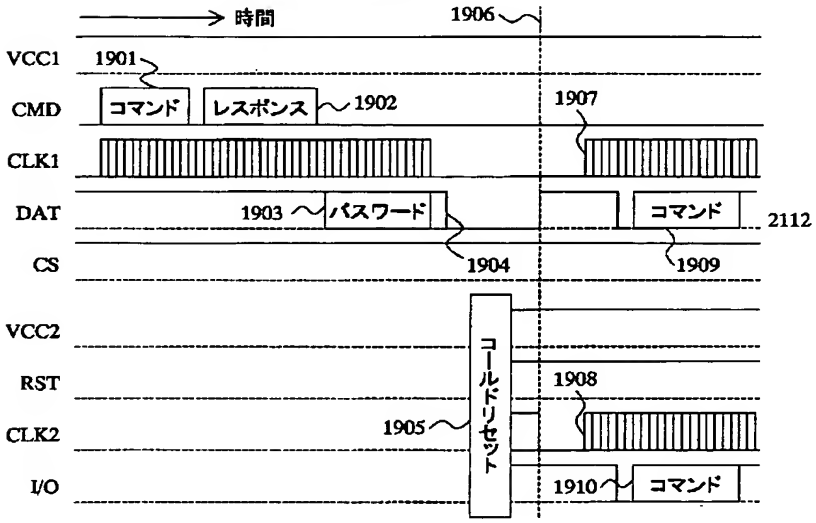
【図１８】

図18



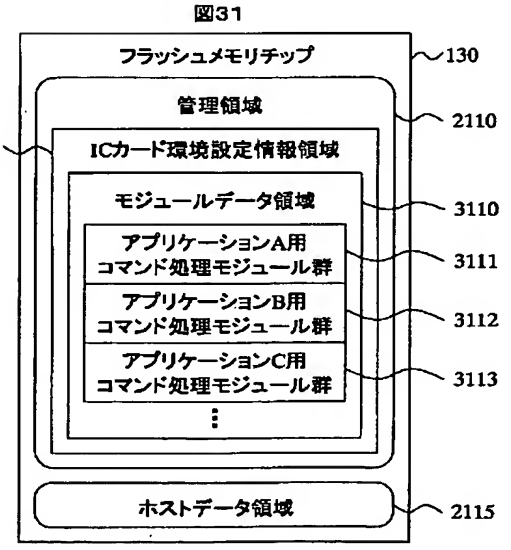
【図１９】

図19



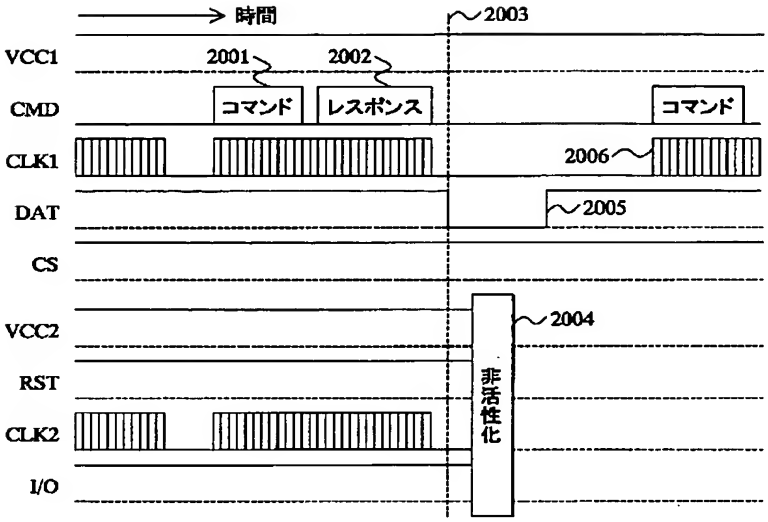
【図３１】

図31



【図２０】

図20



【図３２】

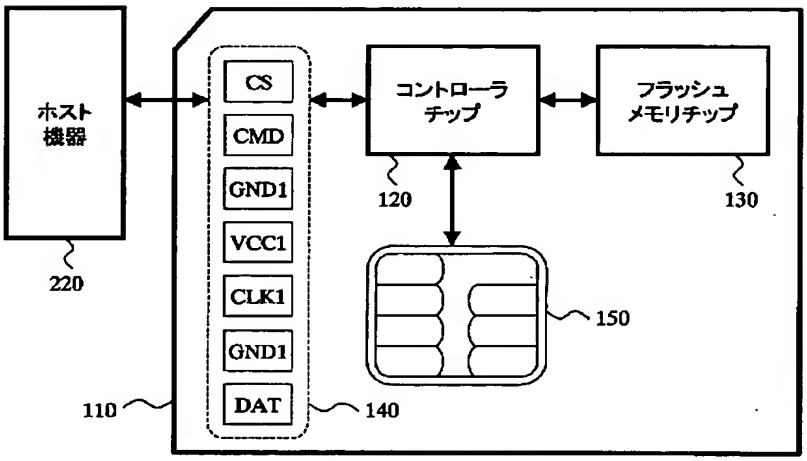
図32

アプリケーションA用
コマンド処理モジュール群　3111

番号	コマンド処理モジュール	
1	第1PIN検証処理モジュール	3201
2	署名検証処理モジュール	3202
3	乱数発生処理モジュール	3203
4	暗号化処理モジュール	3204
5	復号・比較処理モジュール	3205
6	第2PIN検証処理モジュール	3206
7	署名作成処理モジュール	3207

【図２２】

図22



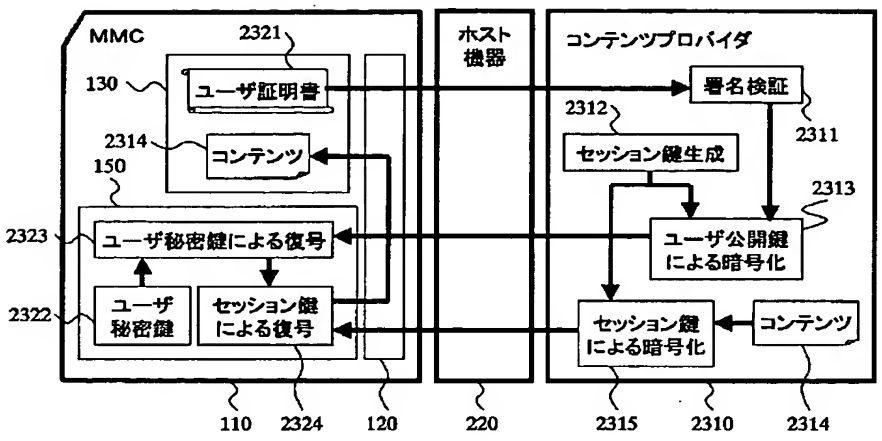
【図３７】

図37

	鍵の名前	説明
3701	モジュール鍵	コマンド処理モジュールにかけられた暗号を解き，実行可能な形式へ復元するための対称鍵。
3702	アプリケーション秘密鍵	アプリケーション発行者が厳重に管理すべき秘密鍵。コマンド処理モジュール更新時に使用。
3703	アプリケーション公開鍵	アプリケーション秘密鍵に対応する公開鍵。ICカードチップ内の各アプリケーションが保持。コマンド処理モジュールロード時，およびコマンド処理モジュール更新時に使用。

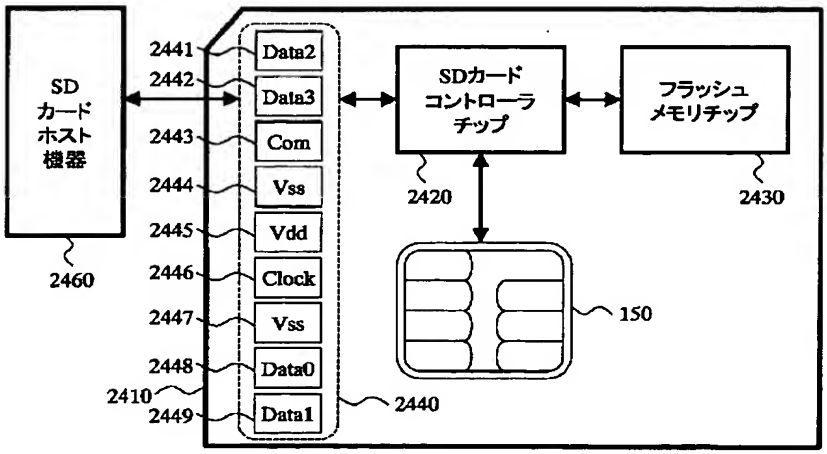
【図２３】

図23



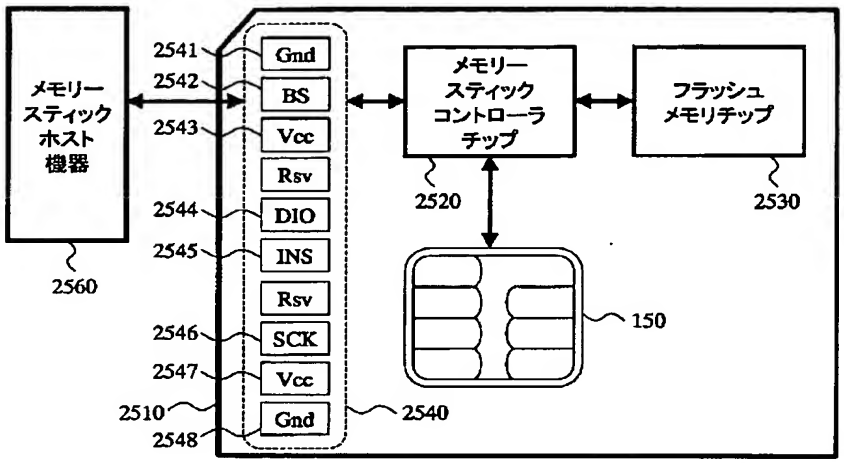
【図２４】

図24



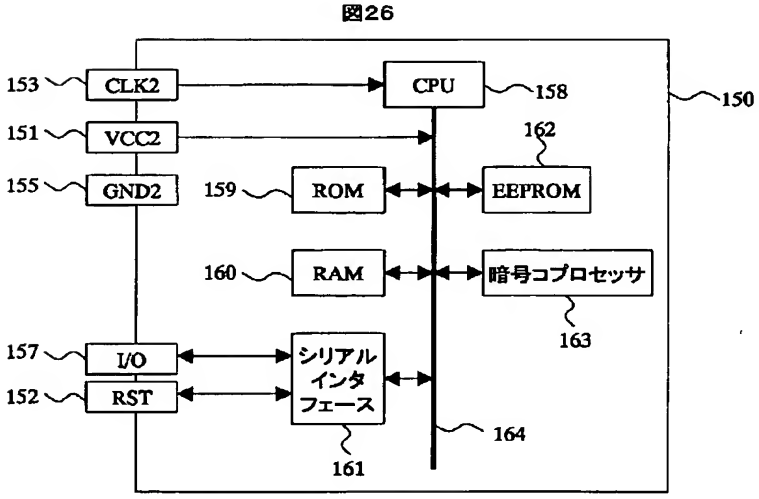
【図２５】

図25



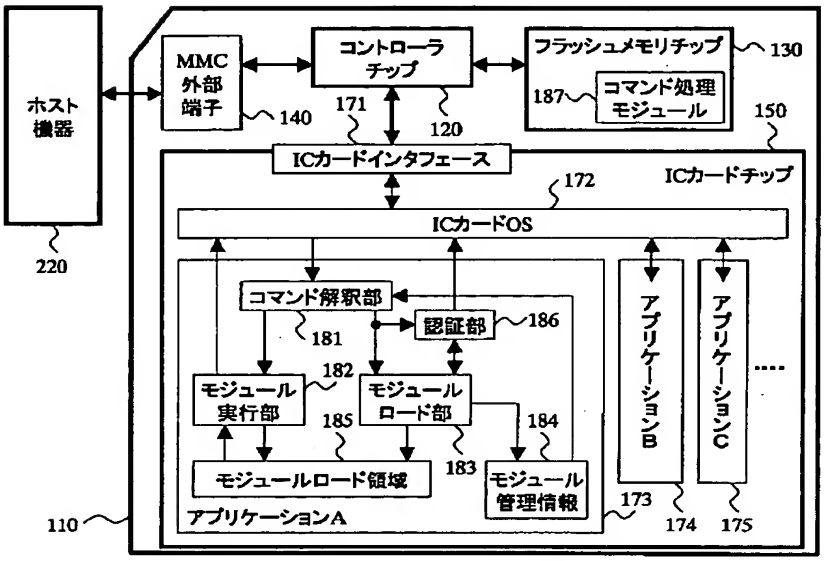
【図２６】

図26



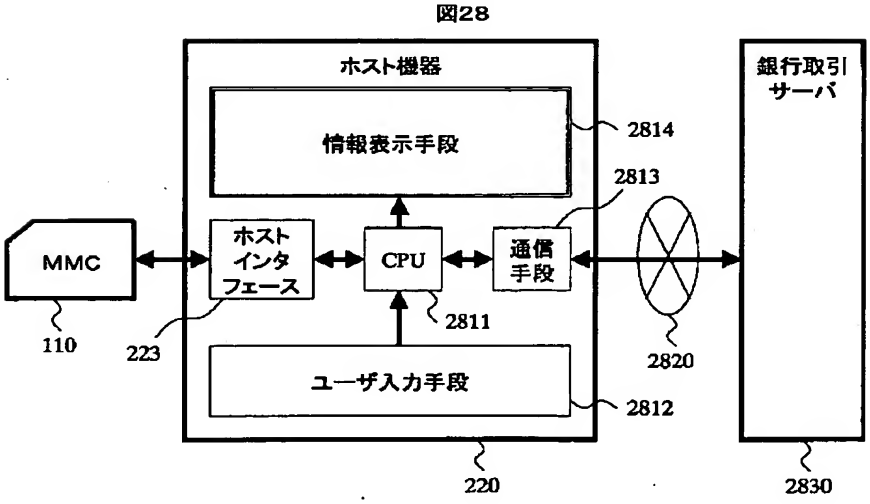
【図２７】

図27



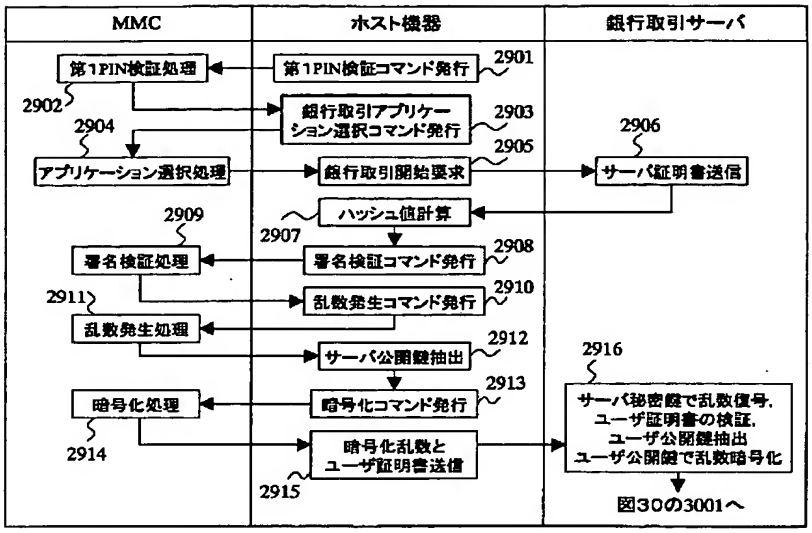
【図２８】

図28



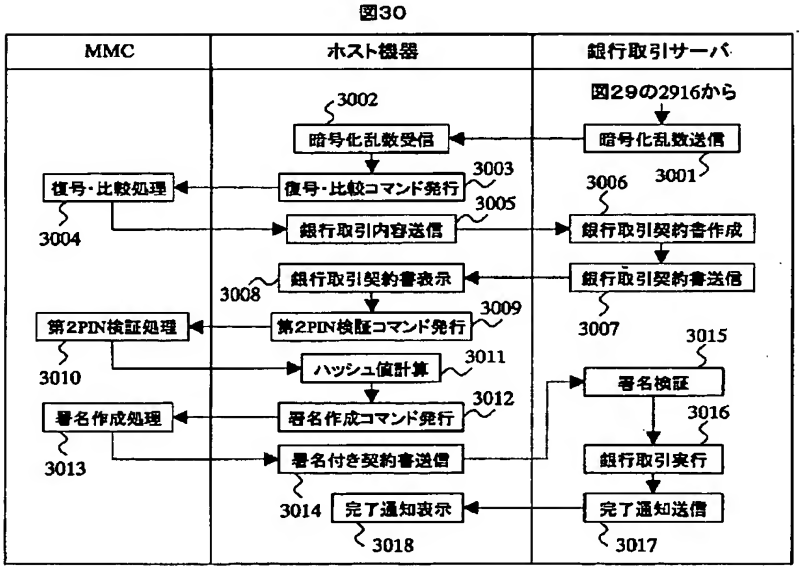
【図２９】

図29



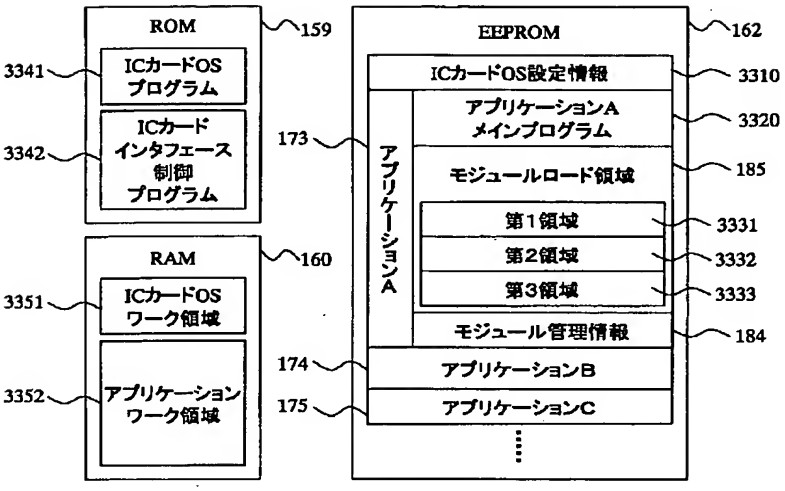
【図３０】

図30



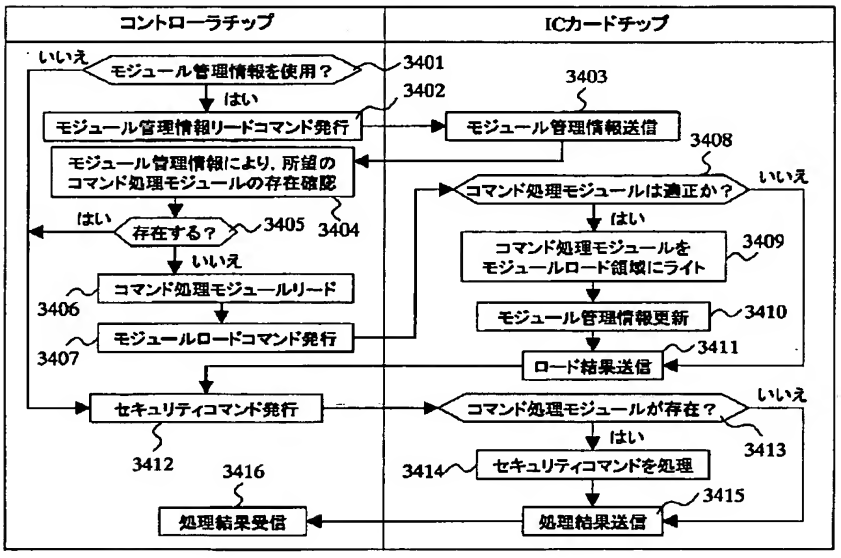
【図３３】

図33



【図３４】

図34



【図３５】

図35

ロード管理情報 3510

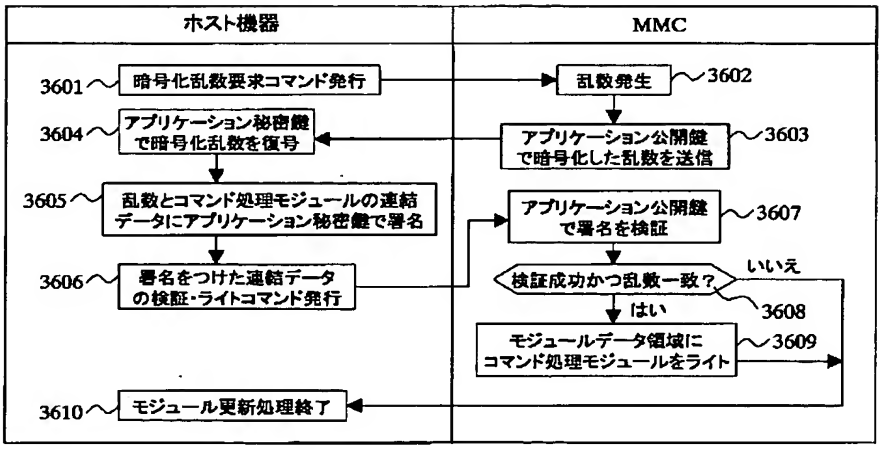
3511 番号	3512 処理内容	3513 サイズ	3514 状態	3515 使用版	3516 最新版	3517 ロード条件
1	第1PIN検証	280	第2領域	3．1	3．1	最新版のみ
2	署名検証	188	第3領域	1．5	2．2	無条件
3	乱数発生	96	なし		1．0	無条件
4	暗号化	256	なし		1．2	最新版のみ
5	復号･比較	388	第1領域	1．4	1．6	1．0以上
6	第2PIN検証	280	なし		3．1	最新版のみ
7	署名作成	176	なし		1．8	1．5以上

ロード領域情報 3520

3522 番号	領域名	サイズ 3521
1	第1領域	400
2	第2領域	320
3	第3領域	200

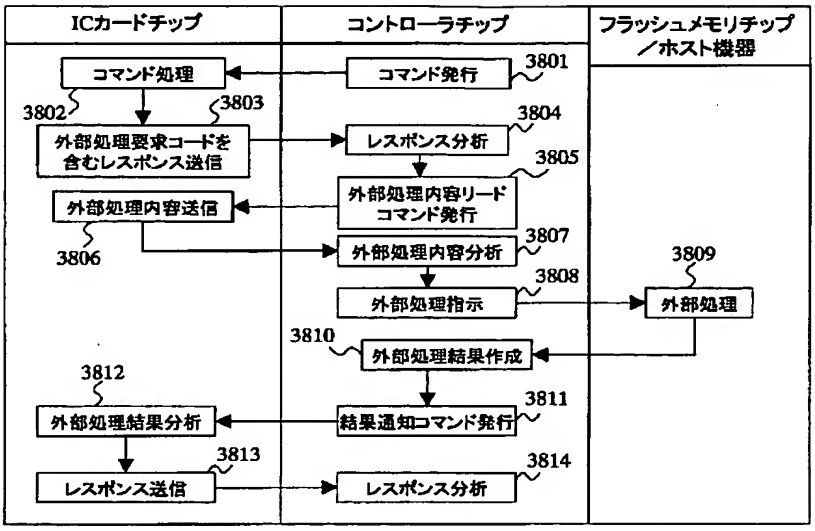
【図３６】

図36



【図３８】

図38



フロントページの続き

(72)発明者　幡野　富久
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内
(72)発明者　片山　国弘
東京都小平市上水本町五丁目20番１号　株式会社日立製作所半導体グループ内
(72)発明者　田中　紀夫
神奈川県川崎市幸区鹿島田890番地　株式会社日立製作所金融システム事業部内
(72)発明者　常広　隆司
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内
(72)発明者　木村　光一
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内

Ｆターム(参考)　2C005 MA04 MB01 MB08 MB10 NA02
NA40 NB01 NB04 SA02 SA03
SA11 SA22
5B035 AA06 AA13 BB09 BC00 CA11